

## [12] 发明专利申请公开说明书

[21] 申请号 01801611.1

[43] 公开日 2002 年 12 月 4 日

[11] 公开号 CN 1383678A

[22] 申请日 2001.4.20 [21] 申请号 01801611.1

[30] 优先权

[32] 2000.4.21 [33] JP [31] 183770/00

[32] 2000.9.5 [33] JP [31] 268042/00

[86] 国际申请 PCT/JP01/03412 2001.4.20

[87] 国际公布 WO01/82605 日 2001.11.1

[85] 进入国家阶段日期 2002.2.5

[71] 申请人 索尼公司

地址 日本东京都

[72] 发明人 加藤元树 浜田俊也

[74] 专利代理机构 北京市柳沈律师事务所

代理人 马莹 邵亚丽

权利要求书 4 页 说明书 63 页 附图 101 页

[54] 发明名称 编码设备和方法、记录介质和程序

[57] 摘要

当作为 AV 流上的属性信息将时间控制标志设置为 1 以便编码和记录 AV 流时, AV 流的流逝时间在预定误差范围内正比于 AV 流的数据字节数。如果将 AV 流文件的时间控制标志设置为 1, 并且如果部分地删除流的对应于一时间的部分, 则在盘上创建对应于该删除时间的空区域, 在该空区域上可以以由流的 TS 平均速率指示的位速率记录数据。

名称	字节数	说明
ChInfo01		
version_number	8*4	bitfield
length	32	uint8_t
Clip_stream_type	8	bitfield
offset_EPM	32	uint8_t
TS_recording_rate	24	uint8_t
reserved	8	bitfield
record_time_and_data	4*14	bitfield
duration	4*8	bitfield
reserved	7	bitfield
time_controlled_flag	1	bitfield
TS_average_rate	24	uint8_t
if (Clip_stream_type == 1) // Bridge-Clip AV stream		
SEPM_start_time_discontinuity	32	uint8_t
etc		
reserved	32	bitfield
reserved_for_system_use	144	bitfield
reserved	11	bitfield
is_format_identifier_valid	1	bitfield
is_original_network_ID_valid	1	bitfield
is_transport_stream_ID_valid	1	bitfield
is_service_ID_valid	1	bitfield
is_country_code_valid	1	bitfield
format_identifier	32	bitfield
original_network_ID	16	uint8_t
transport_stream_ID	16	uint8_t
service_ID	16	uint8_t
country_code	24	bitfield
stream_format_name	16*8	bitfield
reserved_for_system_use	256	bitfield

1. 一种用于编码画面数据的编码设备, 包括:  
编码器, 用于以可变速率编码所述画面数据; 和  
5 控制器, 用于管理控制使得画面编码数据的数量基本上与时间流逝成正比。
2. 根据权利要求 1 所述的编码设备, 其中所述控制器管理控制, 使得如果每个单位时间产生的所述画面编码数据的数量小于预设值, 则将编码填充字节。
- 10 3. 根据权利要求 2 所述的编码设备, 其中所述控制器根据在编码各个画面中产生的数据量校验是否编码填充字节。
4. 根据权利要求 2 所述的编码设备, 其中所述控制器管理编码填充字节的控制, 使得将不在 VBV 缓冲器中产生溢出。
5. 根据权利要求 1 所述的编码设备, 其中所述控制器管理控制, 以其中  
15 所述画面编码数据的数量基本上与时间的流逝成正比的编码模式或以常规编码模式执行编码。
6. 根据权利要求 5 所述的编码设备, 其中所述控制器生成指示编码模式是否是其中所述画面编码数据的数量基本上与时间流逝成正比的这种编码模式的附加信息。
- 20 7. 一种用于编码画面数据的编码方法, 包括  
编码步骤, 用于以可变速率编码所述画面数据; 和  
控制步骤, 用于管理控制使得画面编码数据的数量基本上与时间流逝成正比。
8. 一种记录有用于控制用来编码画面数据的编码设备的计算机可读程  
25 序的记录介质, 所述程序包括:  
编码步骤, 用于以可变速率编码所述画面数据; 和  
控制步骤, 用于管理控制使得画面编码数据的数量基本上与时间流逝成正比。
9. 一种用于使计算机控制用来编码画面数据的编码设备以执行下述步  
30 骤的程序, 包括:  
编码步骤, 用于以可变速率编码所述画面数据; 和

控制步骤，用于管理控制使得画面编码数据的数量基本上与时间流逝成正比。

10. 一种记录有画面数据的记录介质，所述记录介质上记录有包括所述画面数据和与所述画面数据相关的音频数据的 AV 流文件；和

5 指示所述 AV 流文件的记录模式的标志。

11. 根据权利要求 10 的记录介质，其中所述标志是 time-controlled-flag。

12. 根据权利要求 11 的记录介质，其中所述标志指示该模式是在其中进行记录使得文件大小将与从记录的时间开始流逝的时间成正比的模式。

10

权 利 要 求 书  
按照条约第 19 条的修改

---

1. 一种用于编码画面数据的编码设备, 包括:

编码器, 用于以可变速率编码所述画面数据; 和

5 控制器, 具有其中编码的数据量基本上与流逝的时间成正比的编码模式和其中不保证编码的画面数据量与流逝的时间成正比的另一种编码模式, 所述控制器根据有关的编码模式控制编码的画面数据量。

2. 根据权利要求 1 所述的编码设备, 其中所述控制器管理控制, 使得如果每个单位时间产生的所述画面编码数据的数量小于预设值, 则将编码填充字节。

3. 根据权利要求 2 所述的编码设备, 其中所述控制器根据在编码各个画面中产生的数据量校验是否编码填充字节。

4. 根据权利要求 2 所述的编码设备, 其中所述控制器管理编码填充字节的控制, 使得将不在 VBV 缓冲器中产生溢出。

15 5. 根据权利要求 1 所述的编码设备, 其中所述控制器管理控制, 以其中所述生成的画面编码数据的数量在预设误差范围内基本上与时间的流逝成正比的编码模式执行编码。

6. 根据权利要求 5 所述的编码设备, 其中所述控制器生成指示编码模式是否是其中所述画面编码数据的数量基本上与时间流逝成正比的这种编码模式的附加信息。

7. 一种用于编码画面数据的编码方法, 包括

编码步骤, 用于以可变速率编码所述画面数据; 和

25 控制步骤, 具有其中编码的数据量基本上与流逝的时间成正比的编码模式和其中不保证编码的画面数据量与流逝的时间成正比的另一种编码模式, 所述控制步骤根据有关的编码模式控制编码的画面数据量。

8. 一种记录有用于控制用来编码画面数据的编码设备的计算机可读程序的记录介质, 所述程序包括:

编码步骤, 用于以可变速率编码所述画面数据; 和

30 控制步骤, 具有其中编码的数据量基本上与流逝的时间成正比的编码模式和其中不保证编码的画面数据量与流逝的时间成正比的另一种编码模式, 所述控制步骤根据有关的编码模式控制编码的画面数据量。



9. 一种用于使计算机控制用来编码画面数据的编码设备以执行下述步骤的程序, 包括:

编码步骤, 用于以可变速率编码所述画面数据; 和

5 控制步骤, 具有其中编码的数据量基本上与流逝的时间成正比的编码模式和其中不保证编码的画面数据量与流逝的时间成正比的另一种编码模式, 所述控制步骤根据有关的编码模式控制编码的画面数据量。

10. 一种记录有画面数据的记录介质, 所述记录介质上记录有包括所述画面数据和与所述画面数据相关的音频数据的 AV 流文件; 和

10 指示是否已经以其中所述 AV 流的文件大小与时间的流逝基本上成正比的编码模式进行了编码的标志。

11. 根据权利要求 10 的记录介质, 其中所述标志是 time-controlled-flag。

## 编码设备和方法、记录介质和程序

5

## 技术领域

本发明涉及一种编码设备和方法、一种记录介质和程序。更具体地，涉及一种其中将记录在记录介质上的数据内容的管理信息生成(render)用于记录的文件的编码设备和方法、记录介质和程序。

10

## 背景技术

近来已经提出了各种类型的光盘，作为能够从记录设备中移出的记录介质。这些可记录的光盘已经是作为几个GB的大容量介质被提出，并且认为有希望作为用于记录诸如视频信号的AV(视听)信号的介质。在要记录在该可记录光盘上的数字AV信号源(供应源)中，有CS数字卫星广播和BS  
15 数字广播。另外，数字系统的地波电视广播也已经为今后使用而提出。

从这些源提供的数字视频信号在MPEG2(运动图像专家组)系统下被例行地进行画面压缩。在记录设备中，设置了适合于该设备的记录速率。如果数字广播的数字视频信号记录在国内使用的常规画面存储介质中，数字视频信号被首先解码，并且随后进行带宽限制以用于记录。在当然包括MPEG1视  
20 频、MPEG2视频和DV系统的数字记录系统的情况下，数字视频信号被首先解码，并且随后根据适合于随后记录的设备的记录速率的编码系统进行重新编码。

但是，其中所提供的位流在记录之前一次解码并随后进行带宽限制及重新编码的记录系统有变坏的画面质量。在记录画面压缩数字信号中，如果输入数字信号的传输速率小于记录和/或再现设备的记录速率，直接记录所提供的没有进行解码或者重新编码的位流的方法在画面质量上仅仅在很少程度上遭受破坏。但是，如果输入数字信号的传输速率超过记录和/或再现设备的记录速率，则重新编码位流和记录经过重新编码的位流确实是必须的，结果，在记录和/或再现设备中解码之后，传输速率将不高于盘记录速率的上  
30 限。

如果位流在其中输入数字信号的位速率随时间增加或者降低的可变速率

系统中传输, 则适于在缓冲器中暂时存储数据和突发方式记录该数据的盘记录设备与具有由旋转头的固定 rpm (每分钟转速) 强加的固定记录速率的带式记录系统相比, 其记录介质的容量浪费要少。

因此, 可以预计, 在不远的将来, 当数字广播变成主流时, 增加对记录和/或再现设备的需求, 其中广播信号记录为数字信号, DataStreamer 无需进行解码或重新编码, 并且其中盘用作记录介质。

如果记录介质的容量增加, 则可以在记录介质上记录大量的数据 (这里指与节目有关的图像或语音)。因此, 大量的节目记录在一个盘上, 结果用户选择记录在盘上的许多节目中的所期望的节目的操作很复杂。因此, 使用户能够在再现盘时确认记录的数据以便相当容易地选择所期望的节目 (数据) 是必要的。

### 发明的公开

因此, 本发明的目的在于生成记录在用于记录的记录介质上的数据内容的管理信息, 由此记录在记录介质上的数据内容和应答信息可以得到适当地管理。

一方面, 本发明提供了一种用于编码画面数据的编码设备, 包括用于以可变速率编码画面数据的编码器, 和用于管理控制以便画面编码数据的数量与时间流逝基本上成正比的控制器。

控制器可以管理控制, 使得如果每单位时间产生的画面编码数据的数量少于预定值, 则将编码填充 (stuffing) 字节。

控制器可以根据在编码各自画面中产生的数据量校验填充字节是否将被编码。

控制器可以管理编码填充字节的控制, 使得 VBV 缓冲器将不产生溢出。

控制器可以管理控制, 以其中画面编码数据的数量基本上与时间流逝成正比的编码模式或常规 (routine) 编码模式执行编码。

控制器可以产生指示编码模式是否是其中画面编码数据的数量与时间流逝基本上成正比的编码模式的附加信息。

另一方面, 本发明提供了一种用于编码画面数据的编码方法, 包括以可变速率编码画面数据的编码步骤, 和管理控制以便画面编码数据的数量与时间流逝基本上成正比的控制步骤。

再另一方面，本发明提供了一种其上记录了用于控制用来编码画面数据的编码设备的计算机可读程序的记录介质，该程序包括以可变速率编码画面数据的编码步骤，和管理控制以便画面编码数据的数量与时间流逝基本上成正比的控制步骤。

- 5 再另一方面，本发明提供了一种用于使得计算机控制用来编码画面数据的编码设备以执行下述步骤的程序：以可变速率编码画面数据的编码步骤，和管理控制以便画面编码数据的数量与时间流逝基本上成正比的控制步骤。

在根据本发明的编码方法和设备、记录介质和程序中，画面数据以可变速率被编码，和控制得到管理，使得画面编码数据的数量与时间流逝大约成正比。

10 正比。

再另一方面，本发明提供了其上记录有画面数据的记录介质，该记录介质上记录了包括与该画面数据相关的画面数据和音频数据的 AV 流文件，和指示 AV 流文件的记录模式的标志。

该标志可以是 time-controlled-flag (时间-控制-标志)。

- 15 该标志可以指示该模式是这样的一个模式，其中进行记录，使得文件的大小将与从记录的时间开始的时间流逝成正比。

在根据本发明的记录介质中，记录有画面数据、包含与该画面数据相关的音频数据的 AV 流文件、和指示 AV 流文件的记录模式的标志。

- 从阅读如附图所示的本发明的实施例，本发明的其它目的、特点和优点
- 20 将更加明显。

### 附图说明

图 1 表示根据本发明记录和/或再现设备的实施例的结构。

- 图 2 表示在记录介质上通过记录和/或再现设备 1 记录的数据的数据格
- 25 式。

图 3 表示实 PlayList 和虚 PlayList。

图 4A, 4B 和 4C 表示实 PlayList 的创建。

图 5A, 5B 和 5C 表示实 PlayList 的删除。

图 6A 和 6B 表示汇编编辑。

- 30 图 7 表示虚 PlayList 中提供的子路径。

图 8 表示 PlayList 重放序列的改变。

- 图 9 表示 PlayList 上的标记和 Clip 上的标记 (mark)。
- 图 10 表示菜单缩略图 (menu thumbnail)。
- 图 11 表示加到 PlayList 的标记。
- 图 12 表示加到 Clip 的标记。
- 5 图 13 表示在 PlayList、Clip 和缩略图文件之间的关系。
- 图 14 表示目录结构。
- 图 15 表示 infr.dvr 的句法。
- 图 16 表示 DVRVolume 的句法。
- 图 17 表示 ResumeVolume 的句法。
- 10 图 18 表示 UIAppInfoVolume 的句法。
- 图 19 表示字符集值表。
- 图 20 表示 TableOfPlayList 的句法。
- 图 21 表示 TableOfPlayList 的另一个句法。
- 图 22 表示 MakersPrivateData 的句法。
- 15 图 23 表示 xxxx.rpls 和 yyyy.vpls 的句法。
- 图 24A 到 24C 表示 PlayList。
- 图 25 表示 PlayList 的句法。
- 图 26 表示 PlayList-type 表。
- 图 27 表示 UIAppInfoPlayList 的句法。
- 20 图 28A 到 28C 表示图 27 所示的 UIAppInfoPlayList 句法中的标志 (flag)。
- 图 29 表示 PlayItem。
- 图 30 表示 PlayItem。
- 图 31 表示 PlayItem。
- 25 图 32 表示 PlayItem 的句法。
- 图 33 表示 IN-time。
- 图 34 表示 OUT-time。
- 图 35 表示 Connection-Condition 表。
- 图 36A 到 36D 表示 Connection-Condition。
- 30 图 37 表示 BridgeSequenceInfo。
- 图 38 表示 BridgeSequenceInfo 的句法。

- 图 39 表示 SubPlayItem。
- 图 40 表示 SubPlayItem 的句法。
- 图 41 表示 Mark-type 表。
- 图 42 表示 PlayListMark 的句法。
- 5 图 43 表示 Mark-type 表。
- 图 44 表示 Mark-time-stamp。
- 图 45 表示 zzzzz.clip 的句法。
- 图 46 表示 ClipInfo 的句法。
- 图 47 表示 Clip-stream-type 的表。
- 10 图 48 表示 offset-SPN。
- 图 49 表示 offset-SPN。
- 图 50A、50B 表示 STC 域。
- 图 51 表示 STC-Info。
- 图 52 表示 STC-Info 的句法。
- 15 图 53 表示 ProgramInfo。
- 图 54 表示 ProgramInfo 的句法。
- 图 55 表示 VideoCondngInfo 的句法。
- 图 56 表示 Video-format 的表。
- 图 57 表示 frame-rate 的表。
- 20 图 58 表示 display-aspect-ratio 的表。
- 图 59 表示 AudioCondngInfo 的句法。
- 图 60 表示 audio-coding 的表。
- 图 61 表示 audio-component-type 的表。
- 图 62 表示 sampling-frequency 的表。
- 25 图 63 表示 CPI。
- 图 64 表示 CPI。
- 图 65 表示 CPI 的句法。
- 图 66 表示 CPI-type 的表。
- 图 67 表示视频 EP-map。
- 30 图 68 表示 EP-map。
- 图 69 表示 EP-map。

- 图 70 表示 EP\_map 的句法。
- 图 71 表示 EP\_typevalues 的表。
- 图 72 表示 EP\_map\_for\_one\_stream-PID 的句法。
- 图 73 表示 TU\_map。
- 5 图 74 表示 TU\_map 的句法。
- 图 75 表示 ClipMark 的句法。
- 图 76 表示 Mark\_type 的表。
- 图 77 表示 Mark\_type-stamp 的表。
- 图 78 示出 menu.thmb 和 mark.thmb 的句法。
- 10 图 79 示出 thumbnail 的句法。
- 图 80 示出 thumbnail\_picture-format 表。
- 图 81A 和 81B 示出 tn\_block。
- 图 82 示出 DVR MPEG2 传输流的结构。
- 图 83 示出 DVR MPEG2 传输流的记录器模型。
- 15 图 84 示出 DVR MPEG2 传输流的播放器模型。
- 图 85 示出源数据包的句法。
- 图 86 示出 TP\_extra\_header 的句法。
- 图 87 示出允许复制指示符的表。
- 图 88 示出无缝连接。
- 20 图 89 示出无缝连接。
- 图 90 示出无缝连接。
- 图 91 示出无缝连接。
- 图 92 示出无缝连接。
- 图 93 示出音频交迭。
- 25 图 94 示出使用 BridgeSequence 的无缝连接。
- 图 95 示出不使用 BridgeSequence 的无缝连接。
- 图 96 示出 DVR STD 模型。
- 图 97 示出用于解码和显示的时序图。
- 图 98 说明了图 1 的 AV 编码器的操作。
- 30 图 99 是用于说明用记录 AV 流的可变位速率编码视频的操作的流程图。
- 图 100 说明了视频缓冲校验器。

图 101 说明了 VBV (视频缓冲校验器) 控制。

图 102 说明了 VBV 控制。

图 103 显示了控制可变位速率的情况。

图 104 显示了可变位速率的控制的情况。

5 图 105 是用于说明图 99 的步骤 S21 的细节的流程图。

图 106 是用于说明图 106 的步骤 S205 的细节的流程图。

图 107 说明了 AV 流的时间流逝和 AV 流上的数据字节数量之间的关系。

图 108 是说明用可变编码速率编码视频和记录 AV 流的操作的流程图。

图 109 是用于说明图 108 的步骤 S400 的细节的流程图。

10 图 110 是用于说明保证时间流逝和 AV 流的数据字节数量之间的正比关系的编码模式的流程图。

图 111 显示了最小化操作的例子。

图 112 显示了在最小化的情况下, 擦除在 IN-time 之前不需要的流数据的例子。

15 图 113 显示了在最大化的情况下, 擦除在 OUT-time 之后不需要的流数据的例子。

图 114 是用于说明形成 EP-map 的典型操作的流程图。

图 115 说明了一种介质。

## 20 实施本发明的最佳方式

[33] 引用附图, 现在详细地说明本发明的实施例。图 1 表示体现本发明的记录和/或再现设备 1 的典型内部结构。首先, 说明构造为记录从外部输入的信号的记录单元 2 的结构。记录和/或再现设备 1 构造为用模拟或者数字数据供给和记录模拟或数字数据。

25 模拟视频信号和模拟音频信号分别提供给端子 11, 12。输入到端子 11 的视频信号输出到分析单元 14 和 AV 编码器 15。输入到端子 12 的音频信号输出到分析单元 14 和 AV 编码器 15。分析单元 14 从输入的视频和音频信号中提取诸如场景变化的特征点。

[34] [35] 30 AV 编码器 15 编码输入的视频和音频信号以将诸如编码视频流 (V)、编码音频流 (A) 和 AV 同步的系统信息输出给多路复用器 16。

[36] 编码视频流是例如用 MPEG (运动图像专家组) 2 系统编码的视频流, 而



编码音频流是根据 MPEG1 系统编码的音频流，编码音频流是在例如 MPEG1 系统中编码的音频流或者根据杜比 AC3（商标）系统编码的音频流。多路复用器 16 基于输入系统信息多路复用输入视频流和音频流以通过开关 17 将多路复用流输出到多路复用的流分析单元 18 和源打包器（source packetizer）

5 19。

[37] 多路复用流例如是 MPEG-2 传输流或者 MPEG-2 节目流。根据在其上记录该流的记录介质 100 的应用格式，源打包器 19 将输入的多路复用流编码成由源数据包构成的 AV 流。在输出到写单元 22 之前，AV 流在 ECC（错误校正和编码）单元 20 和调制单元 22 中用附带的 ECC 代码和调制进行处理，其然后基于由控制器 23 输出的控制信号写（记录）AV 流。

10

[38] 从数字接口或者数字电视调谐器输入的诸如数字电视广播的传输流输入到端子 13。有两个用于记录输入到端子 13 的传输流的记录系统：一个是透明记录系统，另一个是记录之前是重新编码的系统，重新编码的目的在于降低例如记录位速率。记录系统命令信息从作为用户接口的端子 24 输入到控制器 23。

15

[39] 在输入的传输流的透明记录中，输入到端子 13 的传输流通过开关 17 输出到多路复用流分析单元 18 和源打包器 19。如上述，在记录介质上记录 AV 流的随后处理与编码和记录模拟输入的音频和视频信号的随后处理相同，因此，为了简单起见在此不进行说明。

20 如果输入的传输流被重新编码和随后记录，则输入到端子 13 的传输流馈送到多路分用器 26，它多路分用输入的传输流以提取视频流（V）、音频流（A）和系统信息（S）。

[40] 在通过多路分用器 26 提取的流（信息）中，视频流输出到音频解码器 [41] 27，而音频流和系统信息输出到多路复用器 16。音频解码器 27 解码输入的传输流以将编码的视频流（V）输出到多路复用器 16。

25

[42] 从多路分用器 26 输出的且输入到多路复用器 16 的音频流和系统信息以及由 AV 编码器 15 输出的视频流基于输入系统信息被多路复用，并通过开关 17 作为多路复用流输出到多路复用流分析单元 18 和源打包器 19。如上述，在记录介质上记录 AV 流的随后处理是与编码和记录模拟输入的音频和视频信号的随后处理相同，因此，为了简单起见在此不进行说明。

30

[43] 本实施例的记录和/或再现设备 1 在记录介质 100 上记录 AV 流文件，同

时也记录解释该文件的应用数据库信息。对控制器 23 的输入信息是来自分析单元 14 的运动画面的特征信息，来自多路复用流分析单元 18 的 AV 流的特征信息和从端子 24 输入的用户命令信息。

在 AV 编码器 15 编码视频信号时，从分析单元 14 提供的运动画面的特征信息是由分析单元 14 产生的。分析单元 14 分析输入视频和音频信号的内容以产生与输入的运动画面信号的画面特征 (Clip 标记) 有关的信息。该信息是表示诸如节目开始点、场景变化点、CM 商业广告开始和结束点、输入视频信号中的标题或者幻灯机 (telop) 的特征 Clip 标记点的画面的信息，并且还包括与音频信号的立体音/非立体音转换点和消音部分有关的画面和信息的缩略图 (thumbnail)。

上述画面表示信息通过控制 23 馈送到多路复用器 16。当多路复用由控制器 23 指定为 Clip 标记的编码画面时，多路复用器 16 将用于指定 AV 流上编码画面的信息返回到控制器 23。具体地，该信息是画面的 PTS (显示时间戳) 或者是画面编码版本的 AV 流上的地址信息。控制器 23 存储特征画面的分类和用于指定 AV 流上相互关联的编码画面的信息。

来自多路复用流分析单元 18 的 AV 流的特征信息是与要记录的 AV 流编码信息相关的信息，并且通过分析单元 18 记录。例如，特征信息包括 AV 流中 I 画面的时间戳 (time stamp) 和地址信息、系统时钟的非连续点信息、AV 流的编码参数和 AV 流中编码参数的改变点信息。当透明地记录从端子 13 输入的传输流时，多路复用流分析单元 18 从输入传输流中检测前述的 Clip 标记的画面，并且产生用于指定由 Clip 标记和其类型指定的画面的信息。

来自端子 24 的用户指配信息是指定由用户指定的重放域的信息、用于解释重放域内容的字符字母、或者诸如书签或者由用户为他或她喜爱场景而设定的重新开始点的信息。

基于前述输入信息，控制器 23 创建 AV 流数据库 (Clip)，AV 流重放域 (PlayItem) 的组 (PlayList) 的数据库，记录介质 100 记录内容的管理信息 (info.dvr) 和关于缩略图画面的信息。类似于 AV 流，从上述信息构成的应用数据库信息在 ECC 单元 20 和调制单元 21 中进行处理且输入到写单元 22，其然后将数据库文件记录在记录介质 100 上。

随后将详细说明上述的应用数据库信息。

当在记录介质 100 上记录的 AV 流文件 (画面数据和语音数据文件) 和

因此在记录介质 100 上记录的应用数据库信息通过再现单元 3 再现时, 控制器 23 首先命令读出单元 28 从记录介质 100 中读出应用数据库信息。读出单元 28 从记录介质 100 中读出应用数据库信息, 然后从记录介质 100 中读出应用数据库信息以通过由解调单元 29 和 ECC 解码器 30 进行的解调和错误校正处理来将该应用数据库信息发送到控制器 23。

基于应用数据库信息, 控制器 23 将在记录介质 100 上记录的 PlayList 表输出到端子 24 的用户接口。用户从 PlayList 表中选择希望再现的 PlayList。指定为要再现的与 PlayList 相关的信息输入到控制器 23。控制器 23 命令读出单元 28 读出在再现 PlayList 中必需的 AV 流文件。根据该命令, 读出单元 28 从记录介质 100 中读出对应的 AV 流以将所读出的 AV 流输出到解调单元 29。因此, 输入到解调单元 29 的 AV 流通过预置处理被解调和通过 ECC 解码器 30 的处理输出到源解数据包器 (depacketizer) 31。

源解数据包器 31 将从记录介质 100 读出的并以预置方式处理的应用格式的 AV 流变换成可由多路分用器 26 处理的流。多路分用器 26 将形成由控制器 23 指定的 AV 流的重放域 (PlayItem) 的诸如视频流 (V)、音频流 (A) 或者 AV 同步的系统信息 (S) 输出到音频解码器 27, 该 AV 解码器 27 解码视频流和音频流以将重放视频信号和重放音频信号分别输出到相关的端子 32、33。

如果从作为用户接口的端子 24 提供指令进行随机存储重放或者指定重放的信息, 则控制器 23 基于 AV 流数据库 (Clip) 内容确定来自记录介质 100 的 AV 流的读出位置, 以命令读出单元 28 读出 AV 流。如果作为用户选择的 PlayList 将作为从预置时间点进行再现, 则控制器 23 命令读出单元 28 从具有最靠近指定的时间点的时间戳的 I 画面读出数据。

当用户从存储在 Clip 信息的 ClipMark 中的节目的索引点或者场景变化点已经选择了某一 Clip 标记, 即正如作为用户接口所显示的, 当用户从存储在 ClipMark 中的索引点或者场景变化点的缩略图画表中选择了某一画面时, 则控制器 23 确定来自记录介质 100 之 AV 流读出位置, 以命令读出单元 28 读出 AV 流。即, 控制器 23 命令读出单元 28 从具有最靠近已经存储用户选择的画面的 AV 流地址的地址的 I 画面中读出数据。读出单元 28 从指定地址读出数据。读出的数据通过解调单元 29、ECC 解码器 30 和通过源打包器 19 处理以便提供到多路分用器 26, 并且通过音频解码器 27 解码以再现

由标记点画面的地址表示的 AV 数据。

如果用户已经命令快进 (fast forward) 重放, 则控制器 23 命令读出单元 28 基于 AV 流数据库 (Clip) 依次连续地读出 AV 流中的 I 画面数据。

5 读出单元 28 从指定的随机存取点中读出 AV 流的数据。如此读出的数据通过由下游侧的各种元件的处理再现。

现在说明其中用户编辑记录在记录介质 100 上的 AV 流的情况。如果希望指定的记录在记录介质 100 上的 AV 流的重放域, 例如, 如果希望创建从歌曲节目 A 中再现由歌唱者 A 演唱部分并且随后从另一个歌曲节目 B 中再现由同一歌唱者 A 演唱部分的重放例程, 则与重放域的 IN-point (开始点) 和 OUT-point (结束点) 相联系的信息从作为用户接口的端子输入到控制器 23。控制器 23 创建 AV 流重放域 (PlayItem) 的组的数据库 (PlayList)。

15 当用户希望擦除记录在记录介质 100 上的一部分 AV 流时, 与擦除域的 IN-point 和 OUT-point 相关的信息输入到控制器 23, 其然后改进 PlayList 数据库以便仅仅参照需要的 AV 流。控制器 23 还命令写单元 22 擦除 AV 流的不需要的流部分。

现在说明这种情况, 其中用户希望指定记录在记录介质上的 AV 流的重放域以创建新的重放例程 (route) 和以无缝方式互连对应的重放域。在这种情况下, 控制器 23 创建 AV 流重放域 (PlayItem) 的组的数据库 (PlayList) 并进行部分地重新编码与重新多路复用在重放域连接点附近的视频流。

20 在重放域 IN-point 的画面信息和 OUT-point 的画面信息从端子 24 输入到控制器 23。控制器 23 命令读出单元 28 读出所需要的在 IN-point 和在 OUT-point 上再现的画面数据。读出单元 28 从记录介质 100 上读出数据。如此读出的数据通过解调单元 29、ECC 解码器 30 和源打包器 19 输出到多路分用器 26。

25 控制器 23 分析输入到多路分用器 26 的数据以确定对视频流的重新编码方法 (picture-coding-type 的变化和重新编码之编码位数量的分配) 和重新多路复用系统, 以将该系统送到 AV 编码器 15 和多路复用器 16。

30 多路分用器 26 然后将输入的流分离成视频流 (V)、音频流 (A) 和系统信息 (S)。视频流可以分类成输入到音频解码器 27 的数据和输入到多路复用器 16 的数据。前者是重新编码所需要的数据, 并且通过音频解码器 27 解码, 其中具有然后由 AV 编码器 15 重新编码和由此变成视频流的解码画面。

后者数据是从没有重新编码的原始流中拷贝的数据。音频流和系统信息直接输入到多路复用器 16。

多路复用器 16 基于从控制器 23 输入的信息多路复用输入流以输出该多路复用流，其被 ECC 单元 20 和调制单元 21 处理以便发送到写单元 22。写单元 22 基于从控制器 23 提供的控制信号将 AV 流记录在记录介质 100 上。

下面解释应用数据库信息和基于该信息的诸如重放和编辑的操作。图 2 表示用于 AV 流管理的应用格式结构，该结构具有两层，即 PlayList 和 Clip。卷 (volume) 信息管理盘中的所有 Clips 和 PlayList。这里，成对的一个 AV 流和其辅助信息认为是一个对象，并叫作 Clip。AV 流文件叫作 Clip AV 流文件，具有叫作 Clip 信息文件的辅助信息。

一个 Clip AV 流文件存储对应于配置成由应用格式指定的结构的 MPEG 2 传输流的数据。一般说来，文件作为字节串进行处理。Clip AV 流文件的内容在时间轴上扩展，其中具有主要以时间基指定的 Clip (I 画面) 入口点。当给定对预置 Clip 的存取点的时间戳时，Clip 信息文件在找到开始在 Clip AV 流文件中读出的数据的地址信息方面是有用的。

参照图 3，现在解释 PlayList，其是为用户从 Clip 中选择希望观看的重放域和方便地编辑重放域而提供的。一个 PlayList 是 Clip 中的一组重放域。预置 Clip 中的一个重放域叫作 PlayItem 并且由时间轴上的一对 IN\_point 和 OUT\_point 表示。这样，PlayList 是由一组复合 PlayItem 形成的。

PlayList 分类成两个类型，一种是实 PlayList，另一种是虚 PlayList。实 PlayList 共同拥有其引用的 Clip 流部分。即，实 PlayList 在盘上占据对应于其引用的 Clip 流部分之数据容量，并且当擦除实 PlayList 时，其引用的 Clip 流部分的数据也被擦除。

虚 PlayList 不是共同拥有 Clip 数据。因此，如果虚 PlayList 改变或者被删除，而 Clip 的内容不会改变。

解释实 PlayList 的编辑。图 4A 表示实 PlayList 的创建，并且如果 AV 流作为新的 Clip 记录，则引用整个 Clip 的实 PlayList 是新的创建操作。

图 4B 表示实 PlayList 的划分，即在希望点上划分实 PlayList 的操作以将实 PlayList 分成两个实 PlayList。当两个节目在通过单个 PlayList 管理的一个 Clip 管理时，并且当用户趋于重新寄存或者重新记录该节目作为单个独立节目时，进行该划分操作。该操作不会导致 Clip 内容的改变，

即对 Clip 本身的划分。

图 4C 表示将两个实 PlayList 组合成一个新的实 PlayList 之操作的实 PlayList 的组合操作。该组合操作是当诸如用户希望重新寄存两个节目为单个程序时完成的。该操作不会导致 Clip 内容的改变，即是将 Clip 本身组合为一个。

图 5A 表示整个实 PlayList 的删除。如果删除整个预置的实 PlayList 的操作，则由所删除的实 PlayList 引用的 Clip 的相关流部分也被删除。

图 5B 表示实 PlayList 的部分删除。如果删除实 PlayList 的期望部分，则相关 PlayItem 改变为仅仅引用所需要的 Clip 流部分。对应的 Clip 流部分被删除。

图 5C 表示实 PlayList 的最小化。其是使与实 PlayList 相关的 PlayItem 仅仅引用虚 PlayList 需要的 Clip 流部分的操作。对应的虚 PlayList 不需要的 Clip 流部分被删除。

如果实 PlayList 通过上述操作改变使得由实 PlayList 引用的 Clip 流部分被删除，则有这种可能性，即出现采用所删除的 Clip 的虚 PlayList，使得在虚 PlayList 中可以产生因所删除的 Clip 导致的问题。

为了防止这种情况发生，则向用户显示这种消息：“如果存在引用实 PlayList 正在引用的 Clip 流部分的虚 PlayList，且实 PlayList 被删除，则虚 PlayList 本身被删除——可以否？”，通过证实或者警告响应用户的删除操作，此后，执行删除处理，或者取消用户命令的对象。或者，完成实 PlayList 的最小化操作以代替删除虚 PlayList。

现在解释虚 PlayList 的操作。如果操作是对虚 PlayList 进行的，则 Clip 的内容不改变。图 6A 和 6B 表示汇编和编辑（IN-OUT 编辑）。这是创建用户希望观看的重放域的 PlayItem 以创建虚 PlayList 的操作。在 PlayItem 之间的无缝连接是由应用格式支持的，如后述。

如果存在两个实 PlayList1、2 和与对应实 PlayList 相联系的 Clip1、2，则用户指定实 PlayList1 中的预置域（从 IN1 到 OUT1 的域：PlayItem1）为重放域，并且作为下一个要显示的域，还指定实 PlayList2 中的预置域（从 IN2 到 OUT2 的域：PlayItem2）为重放域，如图 6A 所示。准备了由 PlayItem1 和 PlayItem2 组成的单个虚 PlayList，如图 6B 所示。

现在解释虚 PlayList 的重新编辑。该重新编辑可以通过交替虚 PlayList

中的 IN-或者 OUT 点将新 PlayItem 插入或者附加到虚 PlayList 以及删除虚 PlayList 中的 PlayItem 来列举。虚 PlayList 本身也可以被删除。

- 图 7 表示对虚 PlayList 的音频转录（后记录）。其是将音频后记录寄存在虚 PlayList 作为子路径的操作。该音频后记录由应用软件支持。附加音频流作为子路径被加到虚 PlayList 主路径的 AV 流。

实 PlayList 和虚 PlayList 的共同点是图 8 所示的改变（移动）PlayList 重放顺序的操作。该操作是盘（卷）中 PlayList 重放顺序的改变并且由在应用格式中定义的 TableOfPlayList 所支持，正如下面引用例如图 20 要说明的。该操作不会导致 Clip 内容的改变。

- 10 现在解释标记（Mark）。标记是为指定 Clip 和 PlayList 中的加亮或者特征时间而提供的，如图 9 所示。加到 Clip 的标记叫做 ClipMark。ClipMark 是例如节目索引点或者场景变化点，用于指定起因于 AV 流中内容的特征场景。ClipMark 是由例如图 1 分析单元 14 产生的。当 PlayList 再现时，可以引用和使用由 PlayList 引用的 Clip 的标记。

- 15 附加到 PlayList 的标记叫做 PlayListMark（播放表标记）。PlayListMark 是例如由用户设置的书签点或者恢复点（resume point）。对 Clip 和对 PlayList 的标记的设置是将表示标记时间点的时间戳加到标记表。另一方面，标记删除就是从标记表中移去标记的时间戳。结果，AV 流不会由标记设置或者标记删除所改变。

- 20 作为 ClipMark 的另一个格式，由 ClipMark 引用的画面可以以 AV 流中的地址基来指定。Clip 上的标记设置就是将表示标记点的画面的地址基信息加到标记表中。另一方面，标记删除就是从标记表中移去表示标记点画面的地址基信息。结果，AV 流不会由标记设置或者标记删除所改变。

- 25 现在解释缩略图。缩略图是加到 Volume（卷）、PlayList 和 Clip 的静止画面。有两种类型的缩略图，其中之一是作为表示内容的代表画面的缩略图。这主要是用在主画面中，为的是让用户用光标（未示出）选择他或她希望观看的内容。另一种缩略图是表示由标记点指向的场景的画面。

- Volume 和对应 PlayList 需要具有代表画面。当盘设置在记录和/或再现设备 1 中的位置时，Volume 的代表画面被预先提出用作初始地展示表示盘内容的静止画面。注意，盘的意思是预先提出作为盘形状的记录介质 100。PlayList 的代表画面被预先提出用作表示 PlayList 内容的静止画面。

作为 PlayList 的代表画面,可以考虑使用 PlayList 的初始画面作为缩略图(代表画面)。但是,在 0 重放时间的引导画面不必要是表示内容的最佳画面。因此,允许用户设置可选择性画面作为 PlayList 的缩略图。两种类型的缩略图,即作为表示 Volume 之代表画面的缩略图和作为表示 PlayList 之代表画面的缩略图,叫作菜单缩略图。由于经常显示菜单缩略图,这些缩略图需要从盘中以升高的速度读出。因此,以单个文件存储该全部菜单缩略图是有效率的。菜单缩略图不必是从卷的运动画面中提取出的画面,但是可以是从个人计算机或者是数字静止摄像机获取的画面,如图 10 所示。

另一方面,Clip 和 PlayList 需要用复合标记来做标记,同时标记点的画面需要被容易地观看,目的是获得标记位置的内容。表示这种标记点的画面叫作标记缩略图。因此,作为标记操作的正本的画面主要是所提取的标记点画面而不是从外部获取的画面。

图 11 表示附加到 PlayList 的标记和标记缩略图之间的关系,同时图 12 表示附加到 Clip 的标记和标记缩略图之间的关系。与菜单缩略图的区别,标记缩略图用在例如用于表示 PlayList 细节的子菜单中,而其不要求在短的存取时间中读出。所以,无论何时要求缩略图,记录和/或再现设备 1 打开文件和读出一部分该文件,同时即使由记录和/或再现设备 1 进行的文件打开和读出一部分文件要占用一些时间,也不会出现任何问题。

为了减少在卷中出现的文件数,最好将整个标记缩略图存储在一个文件中。尽管 PlayList 可以具有一个菜单缩略图和多个标记缩略图,但不要求用户直接地选择 Clip(通常,Clip 是通过 PlayList 选择的),因此没有必要提供菜单缩略图。

图 13 表示菜单缩略图,标记缩略图,PlayList 和 Clip 之间的关系。在菜单缩略图中,文件归档为从一个 PlayList 向另一个提供的菜单缩略图。在菜单缩略图中,文件包含有卷缩略图,其表示在盘上记录的数据内容。在菜单缩略图中,文件归档为从一个 PlayList 到另一个和从一个 Clip 到另一个创建的缩略图。

下面说明 CPI(特征点信息)。CPI 是包含在 Clip 信息文件中的数据,并且主要用于发现 Clip AV 流文件中的数据地址,在该地址,当提供 Clip 存取点的时间戳时开始数据读出。在本实施例中使用两种类型的 CPI,其中之一是 EP-map,另一种是 TU-map。



EP-map 是从基本流和传输流中提取的入口点 (entry point) (EP) 数据的表。其具有用来发现在此开始解码的 AV 流中入口点地点的地址信息。一个 EP 数据由显示时间戳 (PTS) 和与 PTS 相联系的存取单元的 AV 流中的数据地址构成的, 该数据地址与 PTS 配对。

- 5 EP-map 主要用于两个目的。第一, 其用于发现在由 PlayList 的 PTS 引用的存取单元中的 AV 流的数据地址。第二, EP-map 用于快速前进重放或者快速后退重放。在通过记录和/或再现设备 1 记录输入 AV 流时, 如果流的句法能够被分析, 则在盘上创建和记录 EP-map。

- 10 TU-map 具有从通过数字接口输入的传输数据包的到达时间点得出的时间单元 (TU) 数据的表。其提供了基于到达时间的时间 arrival-time-based 和 AV 流中数据地址之间的关系。当记录和/或再现设备 1 记录输入 AV 流并且流的句法不能被分析时, 则在盘上创建和记录 TU-map。

STCInfo 存储 AV 流文件中的不连续点信息, 该 AV 流文件存储了 MPEG-2 传输流。

- 15 当 AV 流具有 STC 的不连续点时, 相同的 PTS 值可以出现在 AV 流文件中。因此, 如果 AV 流中的时间点是根据 PTS 基指定的, 则存取点的 PTS 不足以指定该点。而且, 还要求包含 PTS 的连续 STC 域的索引。在该格式中, 连续 STC 域和其索引被分别叫做 STC 序列和 STC-sequence-id (STC 序列 id)。STC 序列信息是由 Clip 信息文件的 STCInfo 定义的。

- 20 STC-sequence-id 用在 AV 流文件中并且在具有 TU-map 的 AV 流文件中是可选择的。

节目是每个基本流的集合并且共同拥有对这些流进行同步再现的单个系统时间基。

- 25 再现设备 (图 1 的记录和/或再现设备 1) 在其解码之前知道 AV 流的内容是有用的。这些内容包括例如传输音频或者视频基本流的传输数据包的 PID 值, 或者诸如 HDTV 视频或者 MPEG-2 AAC 音频流的视频或者音频元件的类型。该信息对创建用于给用户显示引用 AV 流的 PlayList 内容的屏幕菜单是有用的。其对于设置对应设备之 AV 解码器和多路分用器的初始状态是同样有用的。

- 30 由于这个原因, Clip 信息文件拥有用于说明节目内容的 ProgramInfo。可以发生的是, 节目内容在其中存储了 MPEG-2 传输流的 AV 流文件中应

当可以改变。例如，可以改变传输视频基本流的传输数据包的 PID，或者可以将视频流的元件类型从 SDTV 改变为 HDTV。

ProgramInfo 存储了关于 AV 流文件中节目内容的改变点的信息。其中节目内容保持不变的 AV 流文件的域叫作 program-sequence (节目序列)。

- 5        该节目序列用在具有 EP-map 的 AV 流文件中，并且在具有 TU-map 的 AV 流文件中是可选择的。

本实施例定义了自身编码流格式 (SESF)。该 SESF 用于编码模拟输入信号和用于解码数字输入信号，其随后用于顺序地将解码的信号编码成 MPEG-2 传输流。

- 10        SESF 定义了与 MPEG-2 传输流和 AV 流有关的基本流。当记录和/或再现设备 1 编码和记录 SESF 流时，在盘上创建和记录 EP-map。

数字广播流使用用于在记录介质 100 进行记录的下述系统之一：第一，数字广播流自动解码成 SESF 流。在这种情况下，所记录的流一定要符合 SESF 并且在盘上一定要准备和记录 EP-map。

- 15        另外，形成数字广播流的基本流自动解码成新的基本流，并且重新多路复用成符合流格式的新的传输流，该流格式由用于标准化数字广播流的组织指定的。在这种情况下，在盘上一定要创建和记录 EP-map。

- 20        例如，假设输入流是符合 ISDB (日本数字 BS 的标准名称) 的 MPEG-2 传输流，其中具有包含 HDTV 视频流和 MPEG AAC 音频流的传输流。HDTV 视频流自动解码成 SDTV 视频流，该 SDTV 视频流和原始的 AAC 音频流重新多路复用成 TS。SDTV 流和传输流两者都需要符合 ISDB 格式。

在记录介质 100 上记录数字广播流的另一个系统是进行输入传输流的透明记录，即记录不变的输入传输流，在这种情况下，EP-map 被列出和记录在盘上。

- 25        或者，输入传输流被透明地记录，即输入传输流被不变地记录，在这种情况下，TU-map 被创建和记录在盘上。

下面解释目录和文件。记录和/或再现设备 1 以下描述为 DVR (数字视频记录)。图 14 表示盘上的典型目录结构。DVR 盘的目录可以列举为：包括

- 30        “DVR” 目录的根目录；和包括 “PLAYLIST” 目录、“CLIPINF” 目录、“M2TS” 目录和 “DATA (数据)” 目录的 “DVR” 目录，如图 14 所示。尽管在根目录下可以创建除这些目录之外的其它目录，但这些在本实施例的应用格式中被

忽略。

在“DATA”目录之下，存储有由 DVR 应用格式指定的所有文件和目录。

“DVR”目录包括四个目录。在“PLAYLIST”目录下放置了实 PlayList 和虚 PlayList 的数据库文件。后面的目录可以存在于没有 PlayList 的状态中。

- 5       在“CLIPINF”下放置了 Clip 数据库。该目录也可以存在于没有 AV 流文件的状态中。在“DATA”目录中，存储有诸如数字 TV 广播的数据广播文件。

- “DVR”目录存储了下述文件。即在 DVR 目录下创建的“info.dvr”以存储应用层的综合信息。在 DVR 目录下，一定有单个 info.dvr。假设该文件名称对 info.dvr 是固定的。“menu.thmb”存储了与菜单缩略图有关的信息。10 在 DVR 目录下，一定有 0 或 1 的标记缩略图。假设该文件名称对“menu.thmb”是固定的。如果没有菜单缩略图，该文件可以不存在。

- “mark.thmb”文件存储了与标记缩略图画面有关的信息。在 DVR 目录下，一定有 0 或 1 的标记缩略图。假设该文件名称对“menu.thmb”是固定的。15 如果没有菜单缩略图，该文件可以不存在。

“PLAYLIST”目录存储了两种类型的 PlayList 文件，它们是实 PlayList 和虚 PlayList。“xxxxx.rpls”文件存储了与一个实 PlayList 有关的信息。对于每个实 PlayList 创建一个文件。文件名称是“xxxxx.rpls”，这里“xxxxx”表示从 0 到 9 的五个数值数字。文件扩展名一定是“rpls”。

- 20       “yyyyy.vpls”存储了与一个虚 PlayList 有关的信息。具有文件名称“yyyyy.vpls”的一个文件是从一个虚 PlayList 到另一个被创建的，这里“yyyyy”表示从 0 到 9 的五个数值数字。文件扩展名一定是“vpls”。

“CLIPINF”目录存储了一个与每个 AV 流文件相联系的文件。

- “zzzzz.clps”是对应于一个 AV 流文件 (Clip AV 流文件或者 Bridge-Clip 流文件) 的 Clip 信息文件。文件名称是“zzzzz.clpi”，这里“zzzzz”表示从 0 25 到 9 的五个数值数字。文件扩展名一定是“clpi”。

- “M2TS”目录存储了 AV 流文件。“zzzzz.m2ts”文件是由 DVR 系统操作的 AV 流文件。这是 Clip AV 流文件或者 Bridge-Clip AV 流文件。文件名称是“zzzzz.m2ts”，这里“zzzzz”表示从 0 到 9 的五个数值数字。文件扩展30 名一定是“m2ts”。

“DATA”目录存储了从数据广播中传输的数据。例如该数据可以是 XML

或者 MPEG 文件。

现在解释每个目录（文件）的句法和语义。图 15 表示 “info.dvr” 文件的句法。“info.dvr” 文件是由三个对象构成，即 DVRVolume（）、TableOfPlayLists（）和 MakersPrivateData（）。

- 5       解释图 15 所示的 info.dvr 的句法。根据来自 “info.dvr” 文件的引导字节的相对字节数，TableOfPlayList-Start-address 表示 TableOfPlayLists（）的引导地址。相对字节数是从 0 开始计算的。

- 根据来自 “info.dvr” 文件的引导字节的相对字节数，MakersPrivateData-Start-address 表示 MakersPrivateData（）的引导地址。相对字节数是从 0 开始计算的。Padding-word 与 “info.dvr” 的句法相关地插入。N1 和 N2 是可选择正整数。每个填充字可以假设为可选择的值。

- DVRVolume（）存储了指示卷（盘）内容的信息。图 16 表示 DVRVolume 的句法。现在解释图 16 所示的 DVRVolume（）的句法。Version-number 表示指示 DVRVolume（）的版本号的四个字符字母。Version-number 编码成与 ISO646 相联系的 “0045”。

      长度由 32 位无符号整数表示，其表明从长度字段之后直接到 DVRVolume（）尾端的字节数目。

- ResumeVolume（）记忆了在 Volume 中最后产生的实 PlayList 或者虚 PlayList 的文件名称。但是，当用户已经中断了实 PlayList 或者虚 PlayList 的重放时的重放位置存储在 PlayListMark（）定义的恢复标记中（见图 42 和 43）。

- 图 17 表示 ResumeVolume（）的句法。解释图 17 所示的 ResumeVolume（）的句法。Valid-flag 表示当该 1 位标志分别设置为 1 或者 0 时 resume-PlayList-name 字段是有效的或是无效的。

      resume-PlayList-name 的 10 字节字段表示要恢复的实 PlayList 或者虚 PlayList 的文件名称。

- 图 16 所示的 DVRVolume（）句法中的 UIAppInfoVolume 存储了与 Volume 相关的用户接口应用程序的参数。图 18 表示 UIAppInfoVolume 的句法，现在解释其语义。Character-set 的 8 位字段表示在 Volume-name 字段中编码的字符字母的编码方法。该编码方法对应于图 19 所示的值。

Name-length 的 8 位字段表示在 Volume-name 字段中表示的 Volume 名称的字节长度。Volume-name 字段表示 Volume 的名称。从字段左边开始计数的 Name-length 数的字节数是有效字符数目，并且表示 Volume 的名称。在这些有效字符字母后面的值可以是任何值。

- 5        Volume-protect-flag 是表示卷中的内容是否能够无限制地展示给用户的标志。如果该标志设置为 1，则仅仅在用户已经成功地正确输入 PIN 号（通过口令）的情况下，卷中的内容允许被展示（再现）给用户。如果该标志设置为 0，即使在 PIN 号没有被用户输入的情况下，卷中的内容也允许展示给用户。
- 10        当用户已经将盘插入播放器时，如果该标志已经设置为 0，或者该标志设置为 1 但用户已经成功地正确输入 PIN 号的话，记录和/或再现设备 1 显示盘中的 PlayList 表。在对应 PlayList 之再现的限制是与 Volume-protect-flag 不相关的，并且是由 UIAppInfoVolume 中定义的 playback-control-flag 表示的。
- 15        PIN 是由从 0 到 9 的四个数值数字构成的，其中的每一个都是根据 ISO/IEC646 编码的。ref-thumbnail-index 字段表示加到卷中的缩略图画面的信息。如果 ref-thumbnail-index 字段是除 0xFFFF 之外的值，则缩略图画面被加到卷中。缩略图画面存储在 menu.thumb 文件中。该画面是使用 menu.thumb 文件中的 ref-thumbnail-index 的值引用的。如果
- 20        ref-thumbnail-index 字段是 0xFFFF，则其表示缩略图画面已经被加到卷中。

- 解释图 15 所示的 info.dvr 句法中的 TableOfPlayList ( )。TableOfPlayList ( ) 存储了 PlayList (实 PlayList 和虚 PlayList) 的文件名称。记录在卷中的所有 PlayList 文件都包含在 TableOfPlayList ( )
- 25        中，该 TableOfPlayList ( ) 表示卷中 PlayList 的缺省的重放序列。

图 20 表示 TableOfPlayList ( ) 的句法，现在解释之。TableOfPlayList ( ) 的 version-number 表示四个字符字母，其表示 TableOfPlayList 的版本号。version-number 一定要根据 ISO646 编码成“0045”。

- 长度是无符号的 32 位整数，其表明从长度字段之后直接到
- 30        TableOfPlayList ( ) 尾端的 TableOfPlayList ( ) 的字节数目。
- Number-of-PlayLists 的 16 位字段表示包括 PlayList-file-name 在内的循

环 (for-loop) 的循环数。该数值数字一定要等于记录在卷中的 PlayList 的数目。PlayList-file-name 的 10 字节数值数字表示 PlayList 的文件名称。

图 21 表示 TableOfPlayList ( ) 句法的另一种结构。图 21 所示的句法是由图 20 所示的其中包含 UIAppInfoPlayList 的句法构成的。通过这种包括 UIAppInfoPlayList 的结构, 在读出 TableOfPlayList 时简单地创建菜单画面变成有可能。下面的解释是根据使用图 20 所示的句法进行的。

解释图 15 所示的 info.dvr 中的 MakersPrivateData。提供 MakersPrivateData 是为了允许记录和/或再现设备 1 的制造者将制造者的私有数据插入 MakersPrivateData ( ) 以用于不同公司的特殊应用。每个制造者的私有数据已经被标准化为 maker-ID, 以识别已经定义它的制造者。MakersPrivateData ( ) 可以包含一个或者多个 maker-ID。

如果预置制造者希望插入私有数据, 并且不同制造者的私有数据已经包含在 MakersPrivateData ( ) 中, 则在不擦除预先存在的旧私有数据的情况下, 新私有数据加到 MakersPrivateData ( )。因此, 在本实施例中, 多个制造者的私有数据能够包含在一个 MakersPrivateData ( ) 中。

图 22 表示 MakersPrivateData 的句法。解释图 22 表示的 MakersPrivateData 的句法。TableOfPlayList ( ) 的 version-number 表示四个字符字母, 其表示 TableOfPlayList 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。长度是无符号 32 位整数, 其表明从长度字段之后直接到 MakersPrivateData ( ) 尾端的 TableOfPlayList ( ) 的字节数目。

Mpd\_blocks-start-address 表示根据来自 MakersPrivateData ( ) 引导字节之字节数的第一个 Mpd\_block ( ) 的前端地址。Number-of-maker-entries 是 16 位无代码整数, 其提供 MakersPrivateData ( ) 中包括的制造者私有数据的入口号。在 MakersPrivateData ( ) 中一定不会出现具有相同 maker-ID 值的两个或多个制造者私有数据。

mpd\_blocks-size 是 16 位无符号整数, 其提供以 1024 字节为单位的一个 mpd\_block 大小。例如, 如果 Mpd\_blocks-size=1, 则其表示一个 Mpd\_block 的大小是 1024 字节。Number-of-mpd-block 是 16 位无符号整数, 其提供 MakersPrivateData ( ) 中包含的 mpd\_block 数。maker-ID 是 16 位无符号

整数，其表示已经创建制造者私有数据之 DVR 系统的模型数代码。编码成 maker-ID 的值是由发许可证者指定的。

- maker-mode-code 是 16 位无符号整数，其表示已经创建制造者私有数据之 DVR 系统的模型数代码。编码成 maker-mode-code 的值是由制造者设置的，该制造者已经接收了格式许可。start\_mpd\_block\_number 是 16 位无符号整数，其表示开始制造者私有数据的 mpd\_block\_number 号。制造者私有数据的前端一定要与 mpd\_block 的前端对齐。start\_mpd\_block\_number 对应于 mpd\_block 循环中的变量 j。

- mpd\_length 是 32 位无符号整数，其表示制造者私有数据的大小。
- 10 mpd\_block 是其中存储了制造者的私有数据的区。MakersPrivateData ( ) 中的所有的 mpd\_block 一定要是相同大小。

- 解释实 Playlist 文件和虚 Playlist 文件，换言之，即 xxxxx.rpls 和 yyyyy.vpls。图 23 表示 xxxxx.rpls (实 Playlist) 和 yyyyy.vpls (虚 Playlist) 的句法，它们具有相同的句法结构。每一个 xxxxx.rpls 和 yyyyy.vpls 都是由三个对象构成，即 Playlist ( )、PlaylistMark ( ) 和 MakersPrivateData ( )。
- 15

根据来自 Playlist 文件前端的相对字节数为单位，PlaylistMark\_start\_address 表示 PlaylistMark ( ) 的引导地址。相对字节数是从 0 开始计算的。

- 20 根据来自 Playlist 文件前端的相对字节数为单位，MakersPrivateData\_start\_address 表示 MakersPrivateData ( ) 的引导地址。相对字节数是从 0 开始计算的。

Padding\_word (填充字) 是根据 Playlist 文件的句法被插入的，其中 N1 和 N2 是可选择正整数。每个填充字可以假设为可选择的值。

- 25 尽管其已经简要地解释了，下面仍然解释 Playlist。除 Bridge-Clip 之外的所有 Clips 中的重放域一定要由盘中的所有 Playlist 引用。而且，两个或者多个实 PlayLists 一定不重叠相同 Clip 中由它们的 PlayItem 表示的重放域。

- 参照图 24A、24B 和 24C。对于所有的 Clips，存在对应的实 Playlist，如图 24A 所示。即使在编辑操作已经关闭之后也可看到该规则，如图 24B 所示。因此，所有的 Clip 一定通过引用实 Playlist 之一来看到。
- 30

参照图 24C, 虚 PlayList 的重放域一定包含在重放域和 Bridge-Clip 重放域中。盘中一定不出现不由任何虚 PlayList 引用的 Bridge-Clip。

- 包含 PlayItem 表的实 PlayList 一定不含有 SubPlayItem。虚 PlayList 包含 PlayItem 表, 并且如果包含在 PlayList ( ) 中的 CPI-type 是 EP-map 类型以及 PlayList-type 是 0 (含有视频和音频的 PlayList), 则虚 PlayList 可以包含一个 SubPlayItem。在本实施例的 PlayList ( ) 中, SubPlayItem 仅仅用于音频后记录。由一个虚 PlayList 拥有的 SubPlayItem 数一定是 0 或者 1。

下面解释 PlayList。图 25 表示现在要解释的 PlayList 句法。

- 10 version-number 表示四个字符字母, 其表示 PlayList ( ) 的版本号。version-number 根据 ISO646 编码成 “0045”。长度是无符号 32 位整数, 其表明从长度字段之后直接到 PlayList ( ) 尾端的 PlayList ( ) 的总字节数目。PlayList-type 是 8 位字段, 其表示 PlayList 类型, 图 26 示出其一个例子。

- 15 CPI-type 是一位标志, 其表示由 PlayItem ( ) 和 SubPlayItem ( ) 引用的 Clip 的 CPI-type 之值。在由一个 PlayList 引用的所有 Clips 的 CPI 中定义的 CPI-type 一定具有相同值。Number-of-PlayItems 是 16 位字段, 其表示出现在 PlayList 中的 PlayItem 数。

- 对应于预置 PlayItem ( ) 的 PlayItem-id 是由其中 PlayItem ( ) 出现在包含 PlayItem ( ) 之循环中的序列定义的。PlayItem-id 以 0 开始。Number-of-SubPlayItems 是 16 位字段, 其表示在 PlayList 中的 SubPlayItem 数。该值是 0 或者 1。附加音频流路径 (音频流路径) 是一种子路径类型。

解释图 25 所示的 PlayList 句法的 UIAppInfoPlayList。

- UIAppInfoPlayList 存储了涉及 PlayList 的用户接口应用程序的参数。图 27 表示现在要解释的 UIAppInfoPlayList 的句法。Character-set 是 8 位字段, 其表示用于编码在 PlayList-name 字段中编码的字符字母的方法。该编码方法对应于与图 19 所示的表一致的值。

- Name-length 是 8 位字段, 其表示在 PlayList-name 字段中表示的 PlayList 名称的字节长度。PlayList-name 字段表示 PlayList 名称。从字段左边计数的 Name-length 数的字节数是有效字符数并且表示 PlayList 名称。在这些有效字符字母后面的值可以是任何值。



Record-time-and-date 是 56 位字段，其存储了记录 PlayList 的日期和时间。该字段是二进制编码的十进制（BCD）编码的年/月/日/小时/分钟/秒的 14 个数值数字。例如，2001/12/23: 01: 02: 03 编码成“0x20011223010203”。

- 5       持续时间（duration）是 24 位字段，其表示以小时/分钟/秒为单位的 PlayList 的总重放时间。该字段是二进制编码的十进制（BCD）编码的 6 个数值数字。例如，01: 45: 30 编码成“0x014530”。

- Valid-period 是 32 位字段，其表示 PlayList 的有效时间周期。该字段是 4 位二进制编码的十进制（BCD）编码的 8 个数值数字。Valid-period  
10   用在记录和/或再现设备 1 中，即，当有效周期已经消失的 PlayList 将自动擦除时，例如，2001/05/07 编码成“0x20010507”。

- Maker-ID 是 16 位无符号整数，其表示是最近更新其 PlayList 的 DVR 播放器（记录和/或再现设备 1）的制造者。编码成 Maker-ID 的值分配给 DVD 格式的发放许可证者。Maker-code 是 16 位无符号整数，其表示是最近更新  
15   的 PlayList 的 DVR 播放器的模型数。编码成 Maker-code 的值是由制造者确定的，该制造者已经接收 DVR 格式的许可。

如果 playback-control-flag 的标志设置为 1，则其 PlayList 仅仅当用户成功地输入 PIN 号时再现。如果该标志设置为 0，则用户在不输入 PIN 号的情况下可以观看该 PlayList。

- 20       如果 write-protect-flag 设置为 1，则除 write-protect-flag 之外，该 PlayList 的内容既不能被擦除也不能改变。如果该标志设置为 0，用户可自由地擦除或者改变该 PlayList。如果该标志设置为 1，则在用户进行擦除、编辑或者重写 PlayList 之前记录和/或再现设备 1 显示请求用户进行重新确认的消息。

- 25       其中 write-protect-flag 设置为 0 的实 PlayList 可以存在，引用实 PlayList 之 Clip 的虚 PlayList 可以存在，并且虚 PlayList 的 write-protect-flag 可以设置为 1。如果用户希望擦除实 PlayList，记录和/或再现设备 1 发出报警给用户以便出现前述虚 PlayList 或者在擦除实 PlayList 之前“最小化”该实 PlayList。

- 30       如果 is-played-flag 设置为 1，如图 28B 所示，则其表示自从其记录以来该 PlayList 至少再现了一次，而如果其设置为 0，则其表示自从其记

录以来该 Playlist 甚至没有再现过一次。

文档 (Archive) 是两位字段, 其表示 Playlist 是原始的还是拷贝的, 如图 28C 所示。ref\_thumbnail\_index 的字段表示代表 Playlist 之缩略图画面的信息。如果 ref\_thumbnail\_index 字段是除 0xFFFF 之外的值, 则代表 Playlist 之缩略图画面被加在 Playlist 中, 其中 Playlist 存储在 menu.thmb 文件中。该画面使用在 menu.thmb 文件中的 Ref\_thumbnail\_index 之值来引用。如果 Ref\_thumbnail\_index 是 0xFFFF, 则没有代表 Playlist 之缩略图画面被加在 Playlist 中。

下面解释 PlayItem。一个 PlayItem() 基本上包含下述数据: 用于指定 Clip 文件名称的 Clip-Information-file-name, 成对指定 Clip 重放域的 IN-time 和 OUT-time, 在 Playlist() 中定义的 CPI-type 是 EP-map 类型的情况下由 IN-time 和 OUT-time 引用的 STC-sequence-id, 以及表示在先 PlayItem 和当前 PlayItem 之连接条件的 Connection-Condition。

如果 Playlist 是由两个或者多个 PlayItem 构成, 则在 Playlist 的全球时间轴上这些 PlayItem 成行排列, 没有时间间隙或者重叠。如果在 Playlist 中定义的 CPI-type 是 EP-map 类型和当前 Playlist 不具有 BridgeSequence(), 则 IN-time 和 OUT-time 对在 STC 连续域上一定表示与 STC-sequence-id 指定时间相同的时间。这种例子示于图 29。

图 30 表示这种情况, 其中由 Playlist() 定义 CPI-type, 如果当前 PlayItem 具有 BridgeSequence(), 则应用现在解释的规则。表示成 IN-time1 的在当前 PlayItem 之前的 PlayItem 之 IN-time 表示在当前 PlayItem 之 BridgeSequenceInfo() 中指定的 Bridge-Clip 的时间。该 OUT-time 一定遵守随后要解释的编码限制。

表示成 IN-time2 的当前 PlayItem 的 IN-time 表示在当前 PlayItem 的 BridgeSequenceInfo() 中指定的 Bridge-Clip 的时间。该 IN-time 也一定遵守后面要解释的编码限制。表示成 OUT-time2 的当前 PlayItem 的 PlayItem 之 OUT-time 表示关于由当前 PlayItem 的 STC-sequence-id 指定的 STC 连续域的时间。

如果 Playlist() 的 CPI-type 是 TU-map 类型, 则成对的 PlayItem 的 IN-time 和 OUT-time 表示关于相同 Clip AV 流的时间, 如图 31 所示。

PlayItem 句法示于图 32。对于图 32 所示的 PlayItem 的句法,

Clip-Information-file-name 的字段表示 Clip 信息的文件名称。由该 Clip 信息文件的 ClipInfo ( ) 定义的 Clip-stream-type 一定表示 Clip AV 流。

STC-sequence-id 是 8 位字段和表示由 PlayItem 引用的连续 STC 域的 STC-sequence-id。如果在 PlayList ( ) 指定的 CPI-type 是 TU\_map 类型，  
5 则该 8 位字段没有意义和设置为 0。IN-time 是 32 位字段并且用于存储 PlayItem 的重放开始时间。IN-time 的语义不同于 PlayList ( ) 中定义的 CPI-type，如图 33 所示。

OUT-time 是 32 位字段并且用于存储 PlayItem 的重放结束时间。  
OUT-time 的语义不同于 PlayList ( ) 中定义的 CPI-type，如图 34 所示。

10 Connection-condition 是 2 位字段，其表示在在先 PlayItem 和当前 PlayItem 之间的连接条件，如图 35 所示。图 36A 到 36D 表示图 35 所示的 Connection-condition 的各种状态。

参照图 37 解释 BridgeSequenceInfo。该 BridgeSequenceInfo 是当前 PlayItem 的辅助信息和包括下述信息。即，BridgeSequenceInfo 包括用于  
15 指定 Bridge-Clip AV 流文件的 Bridge-Clip-Information-file-name 和指定相应的 Clip 信息文件的 Bridge-Clip-Information-file-name (图 45)。

其也是关于由在先 PlayItem 引用的 Clip AV 流的源数据包的地址。该源数据包之后是连接 Bridge-Clip AV 流的第一源数据包。该地址叫作  
RSPN-exit-from-previous-Clip。其也是关于由当前 PlayItem 引用的 Clip AV  
20 流之源数据包的地址。该源数据包之前是连接 Bridge-Clip AV 流文件的最后的源数据包。该地址叫作 RSPN-enter-to-current-Clip。

图 37 中，RSPN-arrival-time-discontinuity 表示 Bridge-Clip AV 流的源数据包地址，其中在到达时间基上没有连续点。该地址被定义在 ClipInfo ( ) 中 (图 46)。

25 图 38 表示 BridgeSequenceInfo 的句法。回到图 38 所示的 BridgeSequenceInfo 的句法，Bridge-Clip-Information-file-name 的字段表示对应于 Bridge-Clip-Information-file 的 Clip 信息文件的文件名称。该 Clip 信息文件的 ClipInfo ( ) 中定义的 Clip-stream-type 一定表示 'Bridge-Clip AV 流'。

30 RSPN-exit-from-previous-Clip 的 32 位字段是关于由在先 PlayItem 引用的 Clip AV 流的源数据包的相对地址。该源数据包之后是连接 Bridge-

Clip AV 流文件的第一源数据包。RSPN\_exit-from-previous-Clip 具有基于源数据包数为单位的大小，并且以 ClipInfo ( ) 中定义的 offset-SPN 值从由在先 PlayItem 引用的 Clip AV 流的第一源数据包开始计数。

- 5 RSPN\_enter-to-current-Clip 的 32 位字段是关于由当前 PlayItem 引用的 Clip AV 流的源数据包的相对地址。该源数据包之前是连接 Bridge-Clip AV 流文件的最后源数据包。RSPN\_enter-to-current-Clip 具有基于源数据包数为单位的大小。RSPN\_enter-to-current-Clip 以 offset-SPN 值作为初始值开始计数，offset-SPN 值是由当前 PlayItem 引用的 Clip AV 流的第一源数据包的 ClipInfo ( ) 中定义的。

- 10 参照图 39 解释 SubPlayItem。仅仅在 PlayList ( ) 的 CPI-type 是 EP-map 类型时才允许使用 SubPlayItem ( )。在本实施例中，SubPlayItem 仅仅用于音频后记录。SubPlayItem ( ) 包括下述数据。第一，其包括用于指定 PlayList 中子路径引用的 Clip 的 Clip-Information-file-name。

- 它还包括用于指定 Clip 中子路径重放域的 SubPath-IN-time 和  
15 SubPath-OUT-time。另外，其包括 sync-PlayItem-id 和用于指定在主路径时间轴上开始子路径再现的时间的 start-PTS-of-PlayItem。由于子路径引用的 Clip AV 流一定不包含 STC 非连续点（系统时间基的非连续点）。用在子路径的 Clip 的音频采样的时钟锁定在主路径的音频采样的时钟。

- 图 40 表示 SubPlayItem 的句法。回到图 40 所示的 SubPlayItem 的句法，  
20 Clip-Information-file-name 的字段表示 Clip 信息文件的文件名称并且由 PlayList 中的子路径使用。该 ClipInfo ( ) 中定义的 Clip-stream-type 一定表示 Clip AV 流。

sync-PlayItem-id 的 8 位字段表示子路径类型。这里，仅仅设置了  
‘0x00’，如图 41 所示，同时其它值被保留为今后使用。

- 25 sync-PlayItem-id 的 8 位字段表示 PlayItem 的 PlayItem-id，PlayItem 包含在主路径时间轴上的子路径的重放开始时间。对应于预置 PlayItem 的 PlayItem-id 的值在 PlayList ( ) 中定义（图 25）。

- sync-start-PTS-of-PlayItem 的 32 位字段表示在主路径时间轴上的子  
路径的重放开始时间，并且表示由 sync-PlayItem-id 引用的 PlayItem 上的  
30 PTS（显示时间戳）的高 32 位。SubPath-IN-time 的高 32 位字段存储了子  
路径的重放开始时间。SubPath-IN-time 表示对应于子路径中第一表示单元

的 33 位 PTS 的高 32 位。

SubPath-OUT-time 的高 32 位字段存储了子路径的重放结束时间。

SubPath-OUT-time 表示由下述方程计算的 Presentation-end-TS 之值的高 32 位。

5       Presentation-end-TS=PTS-OUT+AU-duration.

这里, PTS-OUT 是对应于 SubPath 最后表示单元的 33 位长度的 PTS, AU-duration 是基于 SubPath 最后表示单元的显示周期 90kHz。

下面, 解释图 23 所示的 xxxxx.rpls 和 yyyyy.vpls 句法中的 PlaylistMark ( )。与 Playlist 有关的标记信息被存储该 PlaylistMark 中。

10   图 42 表示 PlaylistMark 的句法。回到图 42 所示的 PlaylistMark 的句法, version-number 是四个字符字母, 其表示该 PlaylistMark ( ) 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。

长度 (Length) 是无符号 32 位整数, 其表示从长度字段之后直接到 PlaylistMark ( ) 尾端的 PlaylistMark ( ) 字节数目。Number-of-PlaylistMarks 是表示 16 位无符号整数, 其表示存储在 PlaylistMark 中的标记数目。  
15   Number-of-PlaylistMarks 可以是 0。Mark-type 是标记类型的 8 位字段并且被编码在图 43 表示的表中。

Mark-time-stamp 的 32 位文件存储表示由标记指定的点的时间戳。  
Mark-time-stamp 的语义不同于在 Playlist ( ) 中定义的 CPI-type, 如图 44  
20   所示。PlayItem-id 是指定 PlayItem 的 8 位字段, 其中输入标记。对应于预置 PlayItem 的 PlayItem-id 的值被定义在 Playlist ( ) 中 (见图 25)。

Character-set 的 8 位字段表示在 mark-name 字段中编码的字符字母的编码方法。编码方法对应于图 19 所示的值。Name-length 的 8 位字段表示在 mark-name 字段所示的标记名称的字节长度。mark-name 字段表示在  
25   mark-name 字段中表明的标记名称。对应于 Name-length 数的从该字段左边开始的字节数是有效字符字母和表示该标记名称。在 mark-name 字段中, 在这些有效字符字母后面的值可以是任意的。

ref-thumbnail-index 的字段表示加到标记的缩略图画面的信息。如果 ref-thumbnail-index 的字段不是 0xFFFF, 则缩略图画面被加到其标记, 同时缩略图画面存储在 mark.thmb 文件中。该画面在 mark.thmb 文件中被引用,  
30   其使用 ref-thumbnail-index 的值, 如后面的解释。如果

ref\_thumbnail\_index 的字段是 0xFFFF, 则其表示没有缩略图画面被加到标记。

现在解释 Clip 信息文件。zzzzz.clpi (Clip 信息文件) 是由六个对象构成的, 如图 45 所示。这些是 ClipInfo ( )、STC-Info ( )、Program ( )、CPI ( )、ClipMark ( ) 和 MarkersPrivateData ( )。对于 AV 流 (Clip AV 流或者 Bridge-Clip AV 流) 和对应的 Clip 信息文件, 使用相同的 “zzzzz” 数字串。

解释回到图 45 所示的 zzzzz.clpi (Clip 信息文件) 的句法。

ClipInfo-Start-address 表示 ClipInfo ( ) 的前端地址, 其中具有从 zzzzz.clpi 文件的前端字节作为单位的相对字节数。该相对字节数是从 0 计数。

STC-Info-Start-address 表示 STC-Info 的前端地址, 其中具有从 zzzzz.clpi 文件的前端字节作为单位的相对字节数。

ProgramInfo-Start-address 表示 ProgramInfo ( ) 的前端地址, 其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。CPI-Start-address 表示 CPI ( ) 的前端地址, 其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。

ClipMark-Start-address 表示 ClipMark ( ) 的前端地址, 其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。MarkersPrivateData-Start-address 表示 MarkersPrivateData ( ) 的前端地址, 其中具有从 zzzzz.clpi 文件之前端字节作为单位的相对字节数。该相对字节数是从 0 计数。Padding-word 根据 zzzzz.clpi 文件的句法被插入。N1, N2, N3, N4 和 N5 一定是 0 或者是可选择正整数。对应的填充字也可以假设为可选择值。

现在解释 ClipInfo。图 46 表示 ClipInfo 的句法。在 ClipInfo ( ) 中存储了对应的 AV 流文件的属性信息 (Clip AV 流或者 Bridge-Clip AV 流文件)。

回到图 46 所示的 ClipInfo 的句法, version-number 是四个字符字母, 其表示该 ClipInfo ( ) 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。长度是无符号 32 位整数, 其表明从长度字段之后直接到 ClipInfo ( ) 尾端的 ClipInfo ( ) 的字节数目。Clip-stream-type 的 8 位字段表示

对应于 Clip 信息文件的 AV 流的类型，如图 47 所示。下面解释对应 AV 流的流类型。

Offset-SPN 的 32 位字段给出 AV 流 (Clip AV 流或者 Bridge-Clip AV 流) 第一源数据包的第一源数据包的源数据包的偏移值。当 AV 流文件首先  
5 记录在盘上时，该 Offset-SPN 一定为 0。

参照图 48，当 AV 流文件的开始部分是通过编辑被擦除时，该 Offset-SPN 可以假定为除 0 之外的值。在本实施例中，引用 Offset-SPN 的相对源数据包数（相对地址）经常以 RSPNxxx 形式说明，这里 xxx 经更新使得 RSPN-xxx 是 RAPN-EP-start。相对源数据包数是以源数据包数作为单位定大小的并且  
10 是从 AV 流文件第一源数据包数开始计数，其中具有 Offset-SPN 值作为初始值。

从 AV 流文件第一源数据包到引用相对源数据包数 (SPN-xxx) 的源数据包的源数据包数是通过下述方程计算的：

$$\text{SPN-xxx} = \text{RSPN-xxx} - \text{Offset-SPN}$$

15 图 48 表示其中 Offset-SPN 为 4 的例子。

TS-recording-rate 是 24 位无符号整数，其为 AV 流所要求的给 DVR 驱动（写单元 22）或者来自 DVR 驱动（读出单元 28）的输入/输出位速率。Record-time-and-date 是 56 位字段，用于存储对应于 Clip 之 AV 流的记录日期和时间，并且以 4 位二进制编码的十进制 (BCD) 进行编码表示成年/月  
20 /日/小时/分钟的 14 个数值数字。例如，2001/2/23: 01: 02: 03 编码成 “0x20011223010203”。

持续时间 (duration) 是 24 位字段，其表示基于到达时钟的以小时/分钟/秒的 Clip 总重放时间。该字段是 4 位二进制编码的十进制 (BCD) 编码的 6 个数值数字。例如，01: 45: 30 编码成 “0x014530”。

25 标志 time-controlled-flag 表示 AV 流文件的记录模式。如果该 time-controlled-flag 是 1，则表示记录模式是这种模式，其中文件大小是正比于自记录以来过去的时间，结果由下述方程所示的条件：

$$\text{Ts-average-rate} * 192 / 188 * (\text{t-start-time}) - \alpha \leq \text{size-clip}(t)$$

$$\leq \text{TS-average-rate} * 192 / 188 * (\text{t-start-time}) + \alpha$$

30 这里 TS-average-rate 是用字节/秒表示的 AV 流文件传输流的平均位速率。

上述方程中， $t$  表示以秒为单位的时间， $\text{start\_time}$  是当 AV 流文件的第一源数据包被记录时的时间点。 $\text{size\_clip}(t)$  是 10\*192 字节， $\alpha$  是依赖于  $\text{TS\_average\_rate}$  的常数。

如果  $\text{time\_controlled\_flag}$  设置为 0，则其表示记录模式没有受到控制，使得记录所过去的时间正比于 AV 流的文件大小。例如，输入传输流以透明方式进行记录。

如果  $\text{time\_controlled\_flag}$  设置为 1， $\text{TS\_average\_rate}$  的 24 位字段表示用在上述方程中的  $\text{TS\_average\_rate}$  的值。如果  $\text{time\_controlled\_flag}$  设置为 0，则该字段没有意义和一定要设置为 0。例如，可变位速率传输流是通过下述顺序编码的：首先，传输速率被设置成  $\text{TS\_recording\_rate}$ 。视频流用可变位速率编码。传输数据包通过不采用空数据包被间断地编码。

$\text{RSPN\_arrival\_time\_discontinuity}$  的 32 位字段是位置的相对地址，在该位置，到达时间基不连续性是在 Bridge-Clip AV 流文件上再现的。 $\text{RSPN\_arrival\_time\_discontinuity}$  是以源数据包数作为单位定大小的，并且从在  $\text{ClipInfo}()$  中定义的作为来自 Bridge-Clip AV 流文件之第一源数据包的  $\text{offset\_SPN}$  的值计数。Bridge-Clip AV 流文件中的绝对地址是基于上述方程计算的：

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

$\text{Reserver\_for\_system\_use}$  的 144 位字段是保留给系统的。如果  $\text{is\_format\_identifier\_valid}$  标志是 1，则其表示  $\text{format\_identifier}$  的字段是有效的。如果  $\text{is\_format\_identifier\_valid}$  标志是 1，则其表示  $\text{format\_identifier}$  字段是有效的。如果  $\text{is\_original\_network\_ID\_valid}$  标志是 1，则其表示  $\text{is\_original\_network\_ID\_valid}$  的字段是有效的。如果标志  $\text{is\_transport\_stream\_ID\_valid}$  是 1，则其表示  $\text{transport\_stream\_ID}$  字节是有效的。如果  $\text{is\_servece\_ID\_valid}$  标志是 1，则其表示  $\text{servece\_ID}$  字段是有效的。

如果  $\text{is\_country\_code\_valid}$  标志是 1，则其表示字段  $\text{country\_code}$  是有效的。 $\text{Format\_identifier}$  的 32 位字段表示由传输流中的注册说明符（ISO/IEC13818-1 中定义的）拥有的  $\text{format\_identifier}$  的值。  
 $\text{original\_network\_ID}$  的 16 位字段表示在传输流中定义的  $\text{original\_network\_ID}$  的值。



Servece-ID 中的 16 位字段表示在传输流中定义的 Servece-ID 的值。  
country-code 的 24 位字段表示由 ISO3166 定义的国家代码。每个字符代码  
被 ISO8859-1 编码。例如, 日本表示成 “JPN” 和编码成 “0x4A0x500x4E”。  
stream-format-name 是 ISO-646 的 15 个字符代码, 其表示提供传输流流定  
5 义之格式组织的名称。该字段中的无效字节具有 “0xFF” 值。

format-identifier、original-network-ID、transport-stream-ID、  
servece-ID, country-code 和 stream-format-name 表示传输流的服务提供  
者。这允许识别对音频或者视频流的编码限制和除音视流或者 SI (服务信  
息) 之外的私有数据的流定义。这些信息能够被用来检查解码器是否能够解  
10 码该流。如果这种解码是可能的, 则该信息可以在开始解码之前用来初始化  
解码器系统。

现在解释 STC-Info。不包含 STC 不连续点 (系统时间基的不连续点)  
的 MPEG-2 传输流中的时间域叫作 STC-sequence。在 Clip 中, STC-sequence  
由 STC-sequence-id 的值指定。图 50A 和 50B 表示连续 STC 域。相同的 STC  
15 值决不出现在相同 STC-sequence 中, 尽管 Clip 的最大时间长度是有限的,  
如后面的解释。因此, 相同 PTS 值也决不出现在相同 STC-sequence 中。如  
果 AV 流包含 N 个 STC 不连续点, 这里  $N > 0$ , 则 Clip 系统时间基被分开成  $(N+1)$   
个 STC-sequence。

STC-Info 存储了地点的地址, 其中产生 STC 了不连续性 (系统时间基  
20 不连续性)。正如参照图 51 的说明, RSPN-STC-start 表示地址, 且在由第  $k+1$   
个 RSPN-STC-start 引用的源数据包之到达时间点的开始和在最后源数据包  
之到达时间点的结束。

图 52 表示 STC-Info 的句法。回到图 52 所示的 STC-Info 的句法,  
version-number 是四个字符字母, 其表示该 STC-Info ( ) 的版本号。  
25 version-number 一定要根据 ISO646 编码成 “0045”。

长度是无符号 32 位整数, 其表明从该长度字段正后面到 STC-Info 尾端  
的 STC-Info ( ) 字节数目。如果 CPI ( ) 的 CPI-type 表示 TU-map 类型, 则  
0 可以设置在该长度字段中。如果 CPI ( ) 的 CPI-type 表示 EP-map 类型,  
则 num-of-STC-sequence 一定是不小于 1 的值。

30 num-of-STC-sequence 的 8 位无符号整数表示 Clip 中的序列数。该值  
表示在该字段后面的循环数。对应于预置 STC-sequence 的 STC-sequence-id

是由次序 (order) 定义的, 该次序中出现对应于包含 RSPN-STC-start 之循环中的 STC-sequence 的 RSPN-STC-start。STC-sequence-id 由 0 开始。

RSPN-STC-start 的 32 位字段表示在此 STC-sequence 在 AV 流文件开始的地址。RSPN-STC-start 表示在 AV 流文件中产生系统时间基不连续性的地址。RSPN-STC-start 也可以是源数据包的相对地址, 源数据包具有 AV 流中新系统时间基的第一个 PCR。RSPN-STC-start 是基于源数据包数的大小, 并且是从具有在 ClipInfo ( ) 中定义的作为初始值之 offset-SPN 的 AV 流文件第一源数据包计数。在该 AV 流文件中, 绝对地址是通过上述方程计算的, 即:

10         $SPN\_xxx = RSPN\_xxx - offset\_SPN$ .

现在参照图 53 解释图 45 所示的 zzzzz.clip 句法中的 ProgramInfo。具有 Clip 中下述特征的时间域叫作 program-sequence。这些特征是: PCR-PID 的值不改变, 音频基本流数也不改变, 对应视频流中的 PID 值不改变, 由其 VideoCodingInfo 定义的编码信息不改变, 音频基本流数也不改变, 对应音频流中的 PID 值不改变, 以及由其 AudioCodingInfo 定义的编码信息不改变。

Program-sequence 在相同时间点上仅仅具有一个系统时间基。Program-sequence 在相同时间点上具有单个 PMT。ProgramInfo ( ) 存储 Program-sequence 开始的地点地址。RSPN\_program-sequence-start 表示地址。

20        图 54 表示 ProgramInfo 的句法。回到图 54 所示的 ProgramInfo, version-number 是四个字符字母, 其表示该 ProgramInfo ( ) 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。

长度是无符号 32 位整数, 其表明从该长度字段正后面到 ProgramInfo 尾端的 ProgramInfo ( ) 字节数目。如果 CPI ( ) 的 CPI-type 表示 TU-map 类型, 则该长度字段可以设置为 0。如果 CPI ( ) 的 CPI-type 表示 EP-map 类型, 则 number-of-program 一定是不小于 1 的值。

number-of-program-sequence 的 8 位无符号整数表示 Clip 中的 program-sequence 数。该值表示在该字段后面的循环数。如果在 Clip 中 program-sequence 不改变, 则 1 一定被设置在 program-sequence 数中。

30        RSPN\_program-sequence-start 的 32 位字段是相对地址, 其中在 AV 流上开始该节目序列。

RSPN-program-sequence-start 是以源数据包数作为单位定大小的, 并且是从 AV 流文件的第一源数据包开始以 ClipInfo ( ) 中定义的 offset-SPN 的值计数。在该 AV 流文件中, 绝对地址是通过下式计算的, 即:

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}.$$

- 5       在循环句法中的 RSPN-program-sequence-start 的值一定要以升序出现。

PCR-PID 的 16 位字段表示传输数据包的 PID, 传输数据包包含对 program-sequence 有效的有效 PCR 字段。Number-of-audios 的 8 位字段表示包含 audio-stream-PID 和 AudioCodingInfo ( ) 之循环数。

- 10      Video-stream-PID 的 16 位字段表示传输数据包的 PID, 传输数据包包含对 program-sequence 有效的视频流。在该字段后面的 VideoCodingInfo ( ) 一定解释由其 Video-stream-PID 引用的视频流的内容。

- 15      Audio-stream-PID 的 16 位字段表示传输数据包的 PID, 传输数据包包含对其 program-sequence 有效的音频流。在该字段后面的 AudioCodingInfo ( ) 一定解释由其 audio-stream-PID 引用的视频流的内容。

其中 Video-stream-PID 的值出现在循环句法中的次序一定等于对 program-sequence 有效的 PMT 中视频流 PID 编码的序列。另外, 其中 audio-stream-PID 的值出现在循环句法中的次序一定等于对 program-sequence 有效的在 PMT 中对音频流 PID 编码的序列。

- 20      图 55 表示图 54 所示的 ProgramInfo 句法中的 VideoCodingInfo 的句法。回到图 55 所示的 VideoCodingInfo 的句法, video-format 的 8 位字段表示对应于 ProgramInfo ( ) 的 Video-stream-PID 的视频格式, 如图 56 所示。

- 25      参照图 57, frame-rate 的 8 位字段表示对应于 ProgramInfo ( ) 中的 video-stream-PID 的视频帧速率。Display-aspect-ratio 的 8 位字段表示对应于 ProgramInfo ( ) 的 video-stream-PID 的视频显示纵横比。

图 59 表示图 54 所示的 ProgramInfo 句法中的 AudioCodingInfo 的句法。回到图 59 所示的 AudioCodingInfo 的句法, audio-format 的 8 位字段表示对应于 ProgramInfo ( ) 的 audio-stream-PID 之音频编码方法, 如图 60 所示。

- 30      audio-component-type 的 8 位字段表示对应于 ProgramInfo ( ) 的 audio-stream-PID 之音频元件类型, 如图 61 所示, 同时 sampling-frequency

的 8 位字段表示对应于 ProgramInfo ( ) 的 audio-stream-PID 之音频采样频率, 如图 62 所示。

解释图 45 所示 zzzzz.clip 句法中的 CPI (特征点信息)。CPI 用于将 AV 流中的时间信息与其文件中的地址相关联。CPI 有两种类型, 即 EP-map 和 TU-map。图 63 中, 如果 CPI ( ) 中的 CPI-type 是 EP-map, 则其 CPI ( ) 包含 EP-map。图 64 中, 如果 CPI ( ) 中的 CPI-type 是 TU-map, 则其 CPI ( ) 包含 TU-map。一个 AV 流具有一个 EP-map 或者一个 TU-map。如果 AV 流是 SESF 传输流, 则对应的 Clip 一定拥有 EP-map。

图 65 表示 CPI 的句法。回到图 65 所示的 CPI 的句法, version-number 是四个字符字母, 其表示该 CPI ( ) 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。长度是无符号 32 位整数, 其表明从该长度字段正后面到 CPI ( ) 尾端的字节数目。CPI-type 是 1 位标志并且表示 Clip 的 CPI 类型, 如图 66 所示。

解释图 65 所示 CPI 句法中的 EP-map。有两种类型 EP-map, 即用于视频流的 EP-map 和用于音频流的 EP-map。EP-map 中的 EP-map-type 区分开这些 EP-map 类型。如果 Clip 包含一个或者多个视频流, 则一定使用用于视频流的 EP-map。如果 Clip 不包含视频流但包含一个或多个音频流, 则一定使用用于音频流的 EP-map。

参照图 67 解释用于视频流的 EP-map。用于视频流的 EP-map 具有数据 stream-PID, PTS-EP-start 和 RSPN-EP-start。stream-PID 表示传输视频流的传输数据包的 PID。PTS-EP-start 表示从视频流序列标头 (header) 开始的存取单元的 PTS。RSPN-EP-start 表示源数据包的地址, 该源数据包包括由 AV 流中的 PTS-EP-start 引用的存取单元的第一字节。

叫做 EP-map-for-one-stream-PID ( ) 的子表是从一个视频流创建的, 该视频流由具有相互相同 PID 的传输数据包传输。如果多个视频流存在于 Clip 中, 则 EP-map 可以包含多个 EP-map-for-one-stream-PID ( )。

用于音频流的 EP-map 具有数据 stream-PID、PTS-EP-start 和 RSPN-EP-start。stream-PID 表示传输音频流的传输数据包的 PID。PTS-EP-start 表示在音频流中的存取单元的 PTS。RSPN-EP-start 表示源数据包的地址, 该源数据包包括由 AV 流的 PTS-EP-start 引用的存取单元的第一字节。

叫做 EP\_map\_for\_one\_stream-PID ( ) 的子表是从一个音频流创建的, 该音频流由具有相互相同 PID 的传输数据包传输。如果多个音频流存在于 Clip 中, 则 EP\_map 可以包含多个 EP\_map\_for\_one\_stream-PID ( )。

回到在 EP\_map 和 STC-Info 之间的关系, 一个

- 5 EP\_map\_for\_one\_stream-PID ( ) 是在与 STC 中的不连续点无关的一个表中创建。将 RSPN-EP-start 的值与 STC-Info ( ) 中定义的 RSPN-STC-start 的值进行比较, 则显示出属于对应 STC-sequence 的 EP\_map 的数据边界 (见图 68)。EP\_map 一定具有对由相同 PID 传输的连续流范围的一个 EP\_map\_for\_one\_stream-PID。在图 69 所示的情况下, 节目#1 和节目#2 具有相同的视频 PID, 但是, 数据范围不是连续的, 使得必须对每个程序一定提供 EP\_map\_for\_one\_stream-PID。
- 10

- 图 70 表示 EP\_map 句法。通过对图 70 所示 EP\_map 句法的解释, EP-type 是 4 位字段并且显示 EP\_map 入口点类型, 如图 71 所示。EP-type 表示在该字段后面的数据字段的语义。如果 Clip 包括一个或多个视频流, 则 EP-type 一定被设置成 0 (‘视频’)。另外, 如果 Clip 不包括视频流但包含一个或多个音频流, 则 EP-type 一定被设置成 1 (‘音频’)。
- 15

- Number-of-stream-PID 的 16 位字段表示具有在 EP\_map 中 Number-of-stream-PID 作为变量的循环的循环的次数。Stream-PID(k) 的 16 位字节表示传输数据包的 PID, 该传输数据包传输由
- 20 EP\_map\_for\_one\_stream-PID(num-EP-entries(k)) 引用的数 k 基本流(视频或者音频流)。如果 EP-type 是 0 (‘视频’), 其基本流一定是视频流。如果 EP-type 等于 1 (‘音频’), 其基本流一定是音频流。

- num-EP-entries(k) 的 16 位字段表示由 EP\_map-entries(k) 引用的 num-EP-entries(k)。EP\_map\_for\_one\_stream-PID-Start\_address(k): 该
- 25 32 位字段表示相对地址位置, 在此, EP\_map\_for\_one\_stream-PID (num-EP-entries(k)) 开始于 EP\_map ( )。该值是由从 EP\_map ( ) 第一字节开始的大小表示的。

Padding-word 一定根据 EP\_map ( ) 句法插入。X 和 Y 一定是可选择性的正整数。对应填充字可以假设任意可选值。

- 30 图 72 表示 EP\_map\_for\_one\_stream-PID 的句法。通过对图 72 所示 EP\_map\_for\_one\_stream-PID 句法的解释, PTS-EP-start 的 32 位字段的语

义不同于 EP-map ( ) 定义的 EP-type。如果 EP-type 等于 0 (‘视频’), 则该字段具有开始于视频流序列标头的存取单元的 33 位精密 PTS 的高 32 位。如果 EP-type 等于 1 (‘音频’), 该该字段具有音频流存取单元的 33 位精密 PTS 的高 32 位。

- 5        RSPN-EP-start 的 32 位字段的语义不同于 EP-map ( ) 定义的 EP-type。如果 EP-type 等于 0 (‘视频’), 则该字段表示源数据包的相对地址, 源数据包包括由 AV 流中 PTS-EP-start 引用的存取单元序列标头的第一字节。另外, 如果 EP-type 等于 1 (‘音频’), 则该字段表示源数据包的相对地址, 源数据包包括由 AV 流中 PTS-EP-start 引用的存取单元音频流中的第一字节。
- 10    节。

RSPN-EP-start 是基于源数据包数作为单位定大小的, 并且是从 AV 流文件的第一源数据包开始以 ClipInfo ( ) 中定义的 offset-SPN 的值为初始值计数。在该 AV 流文件中, 绝对地址是通过下式计算的, 即:

$$\text{SPN-xxx} = \text{RSPN-xxx} - \text{offset-SPN}.$$

- 15        注意, 句法中的 RSPN-EP-start 的值一定以升序出现。

现在参照图 73 解释 TU-map。TU-map 形成基于源数据包到达时钟 (到达时间基的时计) 的时间轴。该时间轴叫做 TU-map-time-axis。

TU-map-time-axis 的原点由 TU-map ( ) 中的 offset-time 表示。

- 20    TU-map-time-axis 从 offset-time 以预置单位被分割, 该单位叫做 time-unit。

在 AV 流的每个 time-unit 中, 第一完全形式之源数据包 AV 流文件上的地址存储在 TU-map 中。这些地址叫做 RSPN-time-unit-start。在 TU-map-time-axis 上开始第 k 个 ( $k \geq 0$ ) time-unit 的时间叫做 TU-start-time(k)。该值是基于下述方程计算的:

- 25         $\text{TU-start-time}(k) = \text{offset-time} + k * \text{time-unit-size}.$

注意, TU-start-time(k) 具有 45kHz 的精确度。

- 图 74 表示 TU-map 的句法。通过解释图 74 所示的 TU-map 的句法, offset-time 的 32 位字段给出与 TU-map-time-axis 相关的偏移时间。该值表示与 Clip 中第一 time-unit 相关的偏移时间。offset-time 的大小是从 27MHz 精确到达时钟作为单位得出的 45kHz 为基础的。如果 AV 流将作为
- 30    新 Clip 进行记录, 则 offset-time 一定设置为 0。

time-unit-size 的 32 位字段提供 time-unit 的大小,并且是以从 27MHz 精确到达时钟作为单位得出的 45kHz 为基础的。最好是, time-unit-size 不长于 1 秒 (time-unit-size $\leq$ 45000)。number-of-time-unit-entries 的 32 位字段表示存储在 TU-map() 中的入口数。

- 5        RSN-time-unit-start 的 32 位字段表示 AV 流中每个 time-unit 开始的位置的相对地址。RSN-time-unit-start 是基于源数据包数作为单位的大小,并且用从 AV 流文件第一源数据包开始的在 ClipInfo() 中定义的 offset-SPN 的值作为初始值计数。AV 流文件中的绝对地址是通过下式计算的:

$$\text{SPN}_{\text{xxx}} = \text{RSPN}_{\text{xxx}} - \text{offset\_SPN}.$$

- 10        注意, 句法循环中的 RSN-time-unit-start 的值一定要以升序出现。如果在数 (k+1) 的 time-unit 中没有源数据包, 则数 (k+1) 的 RSN-time-unit-start 一定等于数 k 的 RSN-time-unit-start。

- 15        通过解释图 45 所示的 zzzzz.clip 句法中的 ClipMark, ClipMark 是与 Clip 有关的标记信息, 并且存储在 ClipMark 中。该标记不是通过用户设置的, 但是是通过记录器 (记录和/或再现设备 1) 设置的。

图 75 表示 ClipMark 的句法。通过解释图 75 所示的 ClipMark 的句法, version-number 是四个字符字母, 其表示该 ClipMark 的版本号。version-number 一定要根据 ISO646 编码成 “0045”。

- 20        长度是无符号 32 位整数, 其表明从长度字段之后直接到 ClipMark() 尾端的 ClipMark() 的字节数目。Number-of-Clip-marks 是 16 位无符号整数, 其表示存储在 ClipMark 中的标记数并且可以等于 0。Mark-type 是 8 位字段, 其表示标记类型和根据图 76 所示的表被编码。

- 25        Mark-time-stamp 是 32 位字段和存储了表示具有专用标记的指针的时间戳。Mark-time-stamp 的语义不同于 PlayList() 中的 CPI-type, 如图 77 所示。

- 30        如果 CPI() 中的 CPI-type 表示 EP-map 类型, 则该 8 位字段表示放置有 Mark-time-stamp 的连续 STC 域的 STC-sequence-id。如果 CPI() 中的 CPI-type 表示 TU-map 类型, 则该 8 位字段没有意义但设置为 0。Character-set 的 8 位字段表示在 mark-name 字段中编码的字符字母的表示方法。该编码方法对应于图 19 所示值。

Name-length 的 8 位字段表示在 mark-name 字段中所示的标记名称的字

节长度。该 mark\_name 字段表示标记名称。对应于 Name-length 数的从该字段左边开始的字节数是有效字符数和表示标记名称。在 mark\_name 字段中, 在这些有效字符字母之后的值可以是任意的。

ref\_thumbnail\_index 的字段表示附加到标记的缩略图画面的信息。

- 5 如果 ref\_thumbnail\_index 字段是不同于 0xFFFF 的值, 该缩略图画面被附加到其标记, 缩略图画面存储在 mark.thumb 文件中。该画面是使用 mark.thumb 文件中的 ref\_thumbnail\_index 之值引用的。如果 ref\_thumbnail\_index 字段是等于 0xFFFF 的值, 则缩略图画面不附加到其标记。

- 10 MakerPrivateData (制造者个人数据) 已经参照图 22 进行了说明, 因此这里不再特别说明。

接下来说明 thumbnail\_information (缩略图信息)。极小画面存储于 menu.thmb (菜单) 文件或 mark.thmb (标记) 文件中。这些文件具有相同的句法结构并拥有唯一的 Thumbnail()。menu.thmb 文件存储表示各自

- 15 PlatyLists 的画面。菜单缩略图的整体存储于唯一的 menu.thmb 文件中。

- mark.thmb 文件存储标记缩略图画面, 它是表示标记点的画面。对应于 PlatyLists 和 Clips 的整体的标记缩略图的整体存储于唯一的 mark.thmb 文件中。由于频繁地增加或删除缩略图, 增加和部分删除的操作必须可以容易并迅速地得到执行。为此, Thumbnail() 具有块结构。画面数据被分为多个部分, 每个部分存储于一个 tn\_block (tn 块) 中。一个画面数据被存储于连续的 tn\_block 中。在 tn\_block 的串中, 可能存在未使用的 tn\_block 块。唯一缩略图画面的字节长度是可变的。
- 20

- 图 78 显示了 menu.thmb 和 mark.thmb 的句法, 并且图 79 显示了图 78 所示的 menu.thmb 和 mark.thmb 的句法结构中的 Thumbnail (缩略图) 的句法。如图 79 所示的 Thumbnail (缩略图) 的句法的说明, version\_number (版本号) 是表示这个 Thumbnail() 的版本号的四个字符的字母。
- 25

version\_number 根据 ISO 646 必须被编码为 “0045”。

- 长度为 32 位的无符号整数, 用于指示从长度字段的尾部到 Thumbnail() 的末端的 MakerPrivateData() 的字节数。以到 Thumbnail() 的前端字节的字节相对数作为单位的话, tu\_block\_start\_address (tu 块开始地址) 是指示第一个 tn\_block 的前端字节地址的 32 位无符号整数。从 0 计数相对字节
- 30



数。Number-of-thumbnails 是 16 位无符号整数,它给出了包含于 Thumbnail() 中的缩略图画面的条目数。

- tu-block-size 是 16 位无符号整数,以 1024 个字节作为单位给出一个 tn-block 的大小。例如,如果 tn-block-size=1,它表示一个 tn-block 的大小是 1024 字节。number-of-tn-blocks 是 116 位的无符号整数,它表示这个 Thumbnail() 中的 tn-block 的条目数。thumbnail-index 是 16 位无符号整数,它表示从 thumbnail-index 字段开始的一个循环的由缩略图信息代表的缩略图画面的索引号。值 0xFFFF 不必用作 thumbnail-index。这个 thumbnail-index 由 UIAppInfoVolume()、UIAppInfoPlaylist()、
- 10 PlaylistMark() 和 ClipMark() 中的 ref-thumbnail-index 参照。

Thumbnail-picture-format 是 8 位无符号整数,代表缩略图画面的格式,并假设图 80 所示的值。在表中,DCF 和 PNG 仅在 menu.thumb 中被允许。标记缩略图必须假设“0x00”值(MPEG-2 视频 1 画面)。

- Picture-data-size 是 32 位无符号整数,以字节为单位表示缩略图画面的字节长度。Start-tn-block-number 是 16 位无符号整数,表示缩略图画面数据开始的 tn-block 的 tn-block 序号。缩略图画面数据的前端必须与 tn-block 的前端相一致。tn-block 序号开始于 0 并相关于 tn-block 的循环中的变量 k 的值。
- 15

- X-picture-length 是 16 位无符号整数,表示缩略图画面的帧的水平方向的像素数。Y-picture-length 是 16 位无符号整数,表示缩略图画面的帧的垂直方向的像素数。tn-block 是其中存储缩略图画面的区域。Thumbnail() 中的所有 tn-block 大小相同(固定长度)并由 tn-block-size 定义大小。
- 20

- 图 81A 和 81B 概要地显示了缩略图画面数据是如何存储在 tn-block 中的。如图 81A 和 81B 所示,如果缩略图画面在 tn-block 的前端开始,并且大小超过 1 tn-block,则它用下一个接着的 tn-block 存储。通过这样做,具有可变长度的数据可以作为固定长度的数据管理,使得可以用更简单的处理进行删除的编辑。
- 25

- 现在说明 AV 流文件。AV 流文件存储于“M2TS”目录(图 14)中。有两种类型的 AV 流文件,即 Clip(剪辑)AV 流文件和 Bridge-Clip(桥式剪辑)AV 流文件。两种 AV 流文件必须是如后面所定义的 DVR MPEG-2 传输流文件的结构。
- 30

首先说明 DVR MPEG-2 传输流。DVR MPEG-2 传输流的结构示于图 82。AV 流文件具有 DVR MPEG-2 传输流的结构。DVR MPEG-2 传输流由整数个联合单元 (Aligned unit) 组成。联合单元的大小是 6144 个字节 (2048\*3 字节)。联合单元从源数据包的第一个字节开始。源数据包 192 个字节长。一个源数据包包括 TP\_extra\_header 和传输数据包。TP\_extra\_header 长度为 4 个字节, 传输数据包 188 个字节长。

一个联合单元由 32 个源数据包组成。DVR MPEG-2 传输流中的最后的联合单元也由 32 个源数据包组成。因此, DVR MPEG-2 传输流在联合单元的边界结束。如果记录在盘上的输入传输流的传输数据包的个数不是 32 的倍数, 则具有空数据包 (PID 的传输数据包=0x1FFFF) 的源数据包必须用作最后的联合单元。文件系统不必使用 DVR MPEG-2 传输流中的额外信息。

图 83 显示了 DVR MPEG-2 传输流的记录器模型。图 83 中所示的记录器是用于规定记录过程的概念性模型。DVR MPEG-2 传输流遵从这个模型。

现在说明 MPEG-2 传输流的输入时序 (timing)。输入的 MPEG-2 传输流是全部传输流或部分传输流。输入的 MPEG-2 传输流必须遵从 ISO/IEC13818-1 或 ISO/IEC 13818-9。MPEG-2 传输流的第  $i$  个字节在时间  $t(i)$  同时输入给 T-STD (ISO/IEC 13818-1 规定的传输流系统目标解码器) 和源分包器。Rpk 是传输数据包的输入速率的瞬时最大值。

27MHz PLL 52 产生 27MHz 的时钟频率。27MHz 的时钟频率被锁定于 MPEG-2 传输流的程序时钟基准 (PCR) 的值。到达时间时钟计数器 53 计数 27MHz 频率的脉冲。Arrival\_time\_clock( $i$ ) 是到达时间时钟计数器在时间  $t(i)$  的计数值。

源分包器 54 将 TP\_extra\_header 附加到传输数据包的整体上以创建源数据包。Arrival\_time\_stamp(到达时间戳) 表示传输数据包的第一个字节到达 T-STD 和源分包器两者的时间。Arrival\_time\_stamp( $k$ ) 是如通过下列方程代表的 Arrival\_time\_clock( $k$ ) 的采样值:

$$\text{Arrival\_time\_stamp}(k) = \text{Arrival\_time\_clock}(k) \% 230$$

这里  $k$  表示传输数据包的第一个字节。

如果两个相邻传输数据包之间的时间间隔是 230/27000000 秒 (大约 40 秒) 或更长, 两个传输数据包的 Arrival\_time\_stamp 的差值应该设置为 230/27000000 秒。为这种情况提供记录器。

平滑缓冲器 55 使输入传输流的位速率平滑。平滑缓冲器不必溢出。Rmax 是当平滑缓冲器不为空时源数据包从平滑缓冲器的输出位速率。如果平滑缓冲器为空，平滑缓冲器的输出位速率为 0。

接下来，说明 DVR MPEG-2 传输流的记录器模型的参数。Rmax 的值由在  
5 与 AV 流文件相关的 ClipInfo() 中定义的 TS-recording-rate (TS 记录速率) 给出。这个值可以从下列方程进行计算：

$$R_{\max} = \text{TS-recording-rate} \times 192 / 188$$

这里 TS-recording-rate 的值是字节大小/秒钟。

如果输入传输流是 SESEF 传输流，则 Rpk 必须等于在与 AV 流文件相关的  
10 ClipInfo() 中定义的 TS-recording-rate。如果输入传输流不是 SESEF 传输流，则基准可以是例如在 MPEG 2 传输流的描述符中定义的值，如用于这个值的 maximum-bitrate-descriptor (最大位速率描述符) 或 partial-stream-descriptor (部分流描述符)。

如果输入传输流是 SESEF 传输流，则平滑缓冲器大小为 0。如果输入传  
15 输流不是 SESEF 传输流，则基准可以是例如在 MPEG 2 传输流的描述符中定义的值，例如在 smoothing-buffer-descriptor (平滑缓冲器描述符) 中、在 short-smoothing-buffer-descriptor (短平滑缓冲器描述符) 中、或在 partial-transport-stream-descriptor (部分传输流描述符) 中定义的值。

对于记录器和播放器 (再现设备)，需要提供足够大小的缓冲器。缺省  
20 的缓冲器大小是 1536 字节。

接下来描述 DVR MPEG 2 传输流的播放器模型。图 84 显示了 DVR MPEG 2 传输流的播放器模型。这是规定再现过程的概念性模型。DVR MPEG 2 传输流遵从这个模型。

27MHz X-tal 61 产生 27Mhz 的频率。27MHz 频率的误差范围必须是+/-  
25 30ppm (27000000+/-810Hz)。到达时间时钟计数器 62 是用于计数 27MHz 频率的脉冲的二进制计数器。Arrival-time-clock(i) 是到达时间时钟计数器在时间 t(i) 的计数值。

在平滑缓冲器 64 中，当平滑缓冲器不为空时，Rmax 是源数据包到平滑缓冲器的输入位速率。如果平滑缓冲器为空，到平滑缓冲器的输入位速率为  
30 0。

借助于所解释的 MPEG 2 传输流的输出时序，如果当前源数据包的

Arrival-time-stamp 等于 Arrival-time-clock(i) 的 LSB (最低有效位) 侧的 30 位, 则从平滑缓冲器移除源数据包的传输数据包。Rpk 是传输数据包速率的瞬时最大值。不允许平滑缓冲器的溢出。

5     DVR MPEG 2 传输流的播放器模型的参数与上述 DVR MPEG 2 传输流的记录器模型的参数相同。

图 85 显示了源数据包的句法结构。Transport-packet() 是在 ISO/IEC 13818-1 中提供的 MPEG 2 传输流。图 85 中所示的源数据包的句法结构中的 TP-Extra-header 的句法结构示于图 86。作为如图 86 所示的 TP-Extra-header 的句法结构的说明, copy-permission-indicator (复制许可指示符) 是代  
10     表传输数据包的有效载荷的复制限制的整数。复制限制可以是自由复制、不再复制、复制一次或禁止复制。图 87 显示了 copy-permission-indicator 的值和它指定的模式之间的关系。

copy-permission-indicator 附加于传输数据包的整体。如果输入传输流使用 IEEE1394 数字接口进行记录, 则 copy-permission-indicator 的值  
15     可以与 EMI (加密模式指示符) 的值相关。如果输入传输流不使用 IEEE1394 数字接口进行记录, 则 copy-permission-indicator 的值可以与嵌入在传输数据包中的 CCI 的值相关。如果模拟信号输入为自编码, 则 copy-permission-indicator 的值可以与模拟信号的 CGMS-A 的值相关。

Arrival-time-stamp 是具有由下列方程中的 Arrival-time-stamp 指  
20     定的值的整数:

$$\text{Arrival-time-stamp}(k) = \text{Arrival-time-clock}(k) \% 230.$$

经由定义的 Clip AV 流, Clip AV 流必须具有如上面描述所定义的 DVR MPEG 2 传输流的结构。在 Clip AV 流中, Arrival-time-clock(i) 必须连续增加。如果在 Clip AV 流中存在系统时间基准 (STC 基准) 的断点, 则 Clip AV  
25     流中的 Arrival-time-clock(i) 必须连续增加。

Clip AV 流的开始和结束之间的 Arrival-time-clock(i) 的最大差值必须是 26 小时。这个限制保证了, 如果在 MPEG 2 传输流的系统时间基准 (STC 基准) 中没有断点, 则相同值的 PTS (表达时间戳) 永远不会出现在 Clip AV 流中。MPEG 2 标准提供了 PTS 具有 233/90000 秒 (大约 26.5 小时) 的环绕  
30     周期。

经由定义 Bridge-Clip AV 流, Bridge-Clip AV 流必须具有如上面描述

所定义的 DVR MPEG 2 传输流的结构。Bridge-Clip AV 流必须包括一个到达时间基准的断点。在到达时间基准的断点的前面和后面的传输流必须遵从编码限制和后面将说明的 DVR-STD。

5 本发明的实施例支持正被编辑的各 PlayItem 之间的视频-音频无缝连接。各 PlayItem 之间的无缝连接保证了对播放器/解码器的“连续数据供应”和“无缝解码处理”。“连续数据供应”是以防止缓冲器下溢所需要的位速率保证供应数据给解码器的能力。当保证数据实时特性时，为了使数据能够从盘读出，数据将以足够大的连续数据块为单位进行存储。

10 “无缝解码处理”意味着播放器在显示记录于盘上的音频视频数据时，解码器的回放输出中不出现停顿或间隔的能力。

现在说明参照无缝连接的 PlayItem 的 AV 流。在先的 PlayItem 和当前的 PlayItem 的无缝显示是否得到保证，可以从当前 PlayItem 中定义的 connection-condition（连接条件）字段进行验证。对于 PlayItem 的无缝连接有两种方法，即使用 Bridge-Clip 的方法和不使用 Bridge-Clip 的方法。

15 图 88 显示了在使用 Bridge-Clip 的情况下，在先的 PlayItem 和当前的 PlayItem 之间的关系。图 88 中，播放器读出的流数据用阴影显示。图 88 中的 TS1 由 Clip1 (Clip AV 流) 的阴影流数据和在 RSPN\_arrival\_time\_discontinuity 之前的阴影流数据组成。

20 TS1 的 Clip1 的阴影流数据是从解码对应于在先的 PlayItem 的 IN\_time（如图 88 中的 IN\_time1 所示）的表示单元所需要的流的地址开始到参照 RSPN\_exit\_from\_previous\_Clip 的源数据包的流数据。在包含于 TS1 的 Bridge-Clip 的 RSPN\_arrival\_time\_discontinuity 之前的阴影数据流是从 Bridge-Clip 的第一个源数据包开始直到参照 RSPN\_arrival\_time\_discontinuity 的源数据包之前一个的源数据包的流数据。

25 在图 88 中，TS2 由 Clip2 (Clip AV 流) 的阴影流数据和接续 Bridge-Clip 的 RSPN\_arrival\_time\_discontinuity 的阴影流数据组成。来自包含于 TS 2 中的 Bridge-Clip 的 RSPN\_arrival\_time\_discontinuity 的阴影流数据是从参照 RSPN\_arrival\_time\_discontinuity 的源数据包开始到 Bridge-Clip 的最后的源数据包的流数据。TS2 的 Clip2 的阴影流数据是参照  
30 RSPN\_enter\_to\_curent\_Clip 的源数据包开始到解码对应于当前 PlayItem 的 OUT\_time（由图 88 中的 OUT\_time2 所示）的表示单元所需要的流的地址

的流数据。

图 89 显示了在不使用 Bridge-Clip 的情况下, 在先的 PlayItem 和当前的 PlayItem 之间的关系。在这种情况下, 由播放器读出的流数据用阴影显示。在图 89 中, TS1 由 Clip1 (Clip AV 流) 的阴影流数据组成。TS1 的 Clip1 的阴影流数据是开始于解码对应于在先的 PlayItem 的 IN-time (如图 89 中的 IN-time1 所示) 的表示单元所需要的流的地址、直到 Clip1 的最后的源数据包的数据。

在图 89 中, TS2 是 Clip2 (Clip AV 流) 的阴影流数据。

TS2 的 Clip2 的阴影流数据是开始于 Clip 2 的第一个源数据包直到解码对应于当前 PlayItem 的 OUT-time (如图 89 中的 OUT-time2 所示) 的表示单元所需要的流的地址的流数据。

在图 88 和 89 中, TS1 和 TS2 是源数据包的连续流。接下来, 详细说明 TS1 和 TS2 的流供应和它们之间的连接条件。首先, 细察无缝连接的编码限制。借助于对传输流的编码结构的限制, 包含于 TS1 和 TS2 中的节目数量必须是 1。包含于 TS1 和 TS2 中的视频流数量必须是 1。包含于 TS1 和 TS2 中的音频流的数量必须是 2 或更少。包含于 TS1 和 TS2 中的音频流的数量必须相等。除上面描述的外, 基本流和私有流也可能包含于 TS1 和/或 TS2 中。

现在说明对视频位流的限制。图 90 显示了由画面显示序列指示的典型无缝连接。为了在结合点附近无缝地演示视频流, 必须通过再编码结合点附近的 Clip 的部分流的处理, 移除显示于 IN-time2 (Clip2 的 IN-time) 前面和 OUT-time1 (Clip1 的 OUT-time) 后面的不需要的画面。

图 91 显示了使用 BridgeSequence (桥接序列) 实现无缝连接的实施例。在 RSPN-arrival-time-discontinuity 之前的 Bridge-Clip 的视频流包括直到对应于图 90 的 Clip1 的 OUT-time1 的画面的编码视频流。这个视频流连接于在先的 Clip1 的视频流, 并被再编码以形成符合 MPEG2 标准的基本流。

在 RSPN-arrival-time-discontinuity 之后的 Bridge-Clip 的视频流包括接续对应于图 90 的 Clip2 的 IN-time2 的画面的编码视频流。这个视频流的解码可以正确地开始, 用于将该视频流连接于下一个 Clip2 视频流。进行再编码以便形成符合 MPEG2 标准的唯一连续基本流。为了创建 Bridge-Clip, 一般需要再编码几个画面, 同时其它画面可以从原始 Clip 中复制。

图 92 显示了不使用图 90 所示的实施例中的 BridgeSequence 而实现无

缝连接的实施例。Clip1 的视频流包括直到对应于图 90 的 OUT-time1 的画面的编码视频流, 并被再编码以给出符合 MPEG2 标准的基本流。用类似的方式, Clip2 的视频流由接续与图 90 的 Clip2 的 IN-time2 相关的画面的编码位流组成。这些编码位流已经被再编码以给出符合 MPEG2 标准的唯一连续基本流。

借助于所说明的视频流的编码限制, TS1 和 TS2 的视频流的帧速率必须相等。TS1 的视频流必须终止于 sequence-end-code(序列结束代码)。TS2 的视频流必须开始于 Sequence header(序列标头)、GOP Header (GOP 标头), 并具有 I 画面。TS2 的视频流必须开始于封闭的 GOP。

- 10 在位流(帧或半帧(field))中定义的视频表示单元必须在结合点处相连续。在结合点处不允许存在半帧或帧的间隔。在使用 3-2 下拉编码的情况下, 重写“top-field-first”和“repeat-first-field”标志是必要的。另外, 可以进行本地再编码以防止产生半帧间隔。

- 15 经由所说明的对音频位流的编码限制, TS1 的音频采样频率和 TS2 的音频采样频率必须相等。TS1 的音频编码方法和 TS2 的音频编码方法(例如 MPEG1 第 2 层、AC-3、SESF、LPCM 和 AAC)必须相等。

- 20 经由所说明的对 MPEG2 传输流的编码限制, TS1 的音频流的最后的音频帧必须包含具有等于 TS1 的最后显示画面的显示结束时间的显示时刻的各音频采样。TS2 的音频流的第一个音频帧必须包含具有等于 TS2 的第一个显示画面的显示开始时间的显示时刻的音频采样。

在结合点处, 不允许在音频表示单元的序列中存在间隔。如图 93 所示, 可能有由小于两个音频帧域的音频表示单元的长度定义的重叠。传送 TS2 的基本流的第一个数据包必须是视频数据包。结合点处的传输流必须遵从后面将要说明的 DVR-STD 标准。

- 25 经由所说明的对 Clip 和 Bridge-Clip 的限制, 不允许在 TS1 或 TS2 中存在到达时间基准的中断。

- 30 下列的限制只应用于使用 Bridge-Clip 的情况。Bridge-Clip AV 流仅在 TS1 的最后源数据包和 TS2 的第一个源数据包的结合点处具有唯一的到达时间基准的断点。ClipInfo() 中定义的 SPN-arrival-time-discontinuity 代表断点的地址, 该地址必须代表参照 TS2 的第一个源数据包的地址。

参照 BridgeSequenceInfo() 中定义的 RSPN-exit-from-previous-Clip

的源数据包可以是 Clip1 中的任何源数据包。该源数据包没有必要位于联合单元的边界。参照 BridgeSequenceInfo() 中定义的 RSPN\_enter-to-current-Clip 的源数据包可以是 Clip2 中的任何源数据包。该源数据包没有必要位于联合单元的边界。

- 5        经由所说明的对 PlayItem 的限制, 在先的 PlayItem 的 OUT-time (如图 89 中所示的 OUT-time1) 必须代表 TS1 的最后视频表示单元的显示结束时间。当前 PlayItem 的 IN-time (如图 89 中所示的 IN-time2) 必须代表 TS2 的第一个表示单元的显示开始时间。

- 10       经由参照图 94 所说明的在使用 Bridge-Clip 的情况下对数据分配的限制, 必须进行无缝连接以保证由文件系统进行的连续数据供应。这必须通过安排 Bridge-Clip AV 流、连接到 Clip1 (Clip 流文件) 和 Clip2 (Clip 流文件) 来实现, 以便满足数据分配的规定。

- 15       必须选择 RSPN\_exit-from-previous-Clip, 以便使得在 RSPN\_exit-from-previous-Clip 之前的 Clip1 (Clip AV 流文件) 的部分流将被安排在不小于半个段 (fragment) 的连续区域内。必须选择 Bridge-Clip AV 流的数据长度, 以便使得该数据将被安排在不小于半个段的连续区域内。必须选择 RSPN\_enter-to-current-Clip, 以使得接续 RSPN\_enter-to-current-Clip 的 Clip2 (Clip AV 流文件) 的部分流将被安排在不小于半个段连续区域内。

- 20       经由参照图 95 所说明的在不使用 Bridge-Clip 的无缝连接的情况下的数据分配限制, 必须进行无缝连接以保证由文件系统进行的连续数据供应。这必须通过安排 Clip1 (Clip AV 流文件) 的最后部分和 Clip2 (Clip AV 流文件) 的第一部分来实现, 以便满足对数据分配的规定。

- 25       Clip1 (Clip AV 流文件) 的最后流部分必须被安排在不小于半个段的连续区域内。Clip2 (Clip AV 流文件) 的第一个流部分必须被安排在不小于半个段的连续区域内。

- 30       接下来说明 DVR-STD。此 DVR-STD 是用于对 DVR MPEG2 传输流的产生和校验中的解码处理进行建模的概念性模型。此 DVR-STD 也是用于对如上所述参照相互无缝连接的两个 PlayItem 的 AV 流的产生和校验中的解码处理进行建模的概念性模型。

图 96 显示了 DVR-STD 模型。作为组成的元件, 图 96 所示的模型包括 DVR



MPEG 2 传输流播放器模型。符号  $n$ 、 $Tb_n$ 、 $Mb_n$ 、 $Eb_n$ 、 $Tbsys$ 、 $Bsys$ 、 $Rxn$ 、 $Rbx_n$ 、 $Rxsys$ 、 $Dn$ 、 $Dsys$ 、 $On$  和  $P9(k)$  与 ISO/IEC 13818-1 的 T-STD 中定义的相同，其中  $n$  是基本流的索引号， $TB_n$  是基本流  $n$  的传输缓冲器。

5  $MB_n$  是基本流  $n$  的多路复用缓冲器并仅为视频流而存在。 $EB_n$  是基本流  $n$  的基本流缓冲器并仅为视频流而存在。 $TBsys$  是用于正在被解码的节目的系统信息的系统目标解码器中的主缓冲器。 $Rxn$  是从  $TB_n$  移除数据的传送速率。 $Rbx_n$  是从  $MB_n$  移除 PES 数据包有效载荷的传送速率并仅为视频流而存在。

10  $Rxsys$  是从  $TBsys$  移除数据的传送速率。 $Dn$  是基本流  $n$  的解码器。 $Dsys$  是与正在被解码的节目的系统信息有关的解码器。 $On$  是视频流  $n$  的再排序缓冲器。 $Pn(k)$  是基本流的第  $k$  表示单元。

现在说明 DVR-STD 的解码过程。在唯一 DVR MPEG 2 传输流正在被再现期间，由源数据包的 `arrival-time-stamp` 确定将传输数据包输入给  $TB_1$ 、 $TB_n$  或  $Tbsys$  的时刻。对  $TB_1$ 、 $MB_1$ 、 $EB_1$ 、 $TB_n$ 、 $Bn$ 、 $Tbsys$  和  $Bsys$  的缓冲操作  
15 的规定与 ISO/IEC 13818-1 提供的 T-STD 中的定义相同，而对决定和显示操作的规定也与 ISO/IEC 13818-1 提供的 T-STD 相同。

现在说明在无缝连接的 Playlist 正在被再现期间的解码处理。这里，说明参照无缝连接的 PlayItem 的两个 AV 流的再现。在下面的说明中，说明例如图 88 中所示的 TS1 和 TS2 的再现。TS1 和 TS2 分别是在先流和当前流。

20 图 97 显示了用于当从给定的 AV 流 (TS1) 向与其无缝连接的下一个 AV 流 (TS2) 转换时进行输入、解码和显示传输数据包的时序图。在从预设的 AV 流 (TS1) 向与其无缝连接的下一个 AV 流 (TS2) 的转换期间，TS2 的到达时间基准的时间轴与 TS1 的到达时间基准的时间轴 (由图 97 中的 ATC1 指示) 不同。

25 再有，TS2 的系统时间基准的时间轴 (由图 97 中的 STC2 指示) 与 TS1 的系统时间基准的时间轴 (由图 97 中的 STC1 指示) 不同。视频显示需要无缝地连续，可是在表示单元的显示尖峰上可能存在交叠。

现在说明对 DVR-STD 的输入时刻。在直到时间  $T1$  的期间，即直到将最后的视频数据包输入到 DVR-STD 的  $TB_1$ ，对 DVR-STD 的  $TB_1$ 、 $TB_n$  或  $TBsys$   
30 的缓冲器的输入时刻由 TS1 的到达时间基准的 `arrival-time-stamp` 确定。

TS1 的剩余数据包必须以 `TS_recording-rate(TS1)` 的位速率被输入  $TB_n$

的缓冲器或 DVR-STD 的 TBSys 中。TS\_recording-rate(TS1)是对应于 Clip1 的 ClipInfo() 中定义的 TS\_recording-rate 的值。TS1 的最后字节被输入缓冲器的时间是时间 T2。因此,在时间 T1 和时间 T2 之间的时间期间,源数据包的 arrival-time-stamp 减小 (discount)。

- 5       如果 N1 是接着 TS1 的最后视频数据包的 TS1 的传输数据包的字节数,则从时间 T1 到时间 T2 的时间 DT1 是 N1 字节以 TS\_recording-rate(TS1) 的位速率完全输入所必需的时间,并根据下列方程计算:

$$DT1=T2-T1=N1/TS\_recording\_rate.$$

- 10       从时间 T1 到时间 T2 的时间期间, RXn 和 Rxsys 两者的值都被变为 TS\_recording-rate(TS1) 的值。除了这条,缓冲操作与 T-STD 的相同。

在时间 T2,到达时间时钟计数器被复位为 TS2 的第一个源数据包的 arrival-time-stamp 的值。对 DVR-STD 的 TB1、TBn 或 TBSys 的缓冲器的输入时刻由 TB2 的源数据包的 arrival-time-stamp 确定。RXn 和 RXsys 两者的值都被变为 T-STD 中定义的值。

- 15       经由所说明的附加音频缓冲和系统数据缓冲,音频解码器和系统解码器除了 T-STD 中定义的缓冲量外,需要具有附加缓冲量(相当于 1 秒钟的数据量),以便允许从时间 T1 到时间 T2 的域的输入数据。

- 20       经由所说明的视频表示时刻,视频表示单元上的显示必须连续地经过结合点,即没有间隔。应注意到,STC1 是 TS1 的系统时间基准的时间轴(如图 97 中的 STC1 所指示),而 STC2 是 TS2 的系统时间基准的时间轴(如图 97 中的 STC2 所指示)。正确地,STC2 开始于 TS2 的第一个 PCR 已经被输入到 T-STD 的时间。

- 25       STC1 和 STC2 之间的偏移如下确定:  $PTS1_{end}$  是对应于 TS2 的最后视频表示单元的 STC1 上的 PTS。 $PTS2_{start}$  是对应于 TS2 的第一个视频表示单元的 STC2 上的 PTS,  $T_{pp}$  是 TS1 的最后视频表示单元的显示时间周期。两个系统时间基准之间的偏移 STC\_delta 根据下列方程计算:  $STC\_delta = PTS1_{end} + T_{pp} - PTS2_{start}$ 。

- 30       经由所说明的音频表示时刻,在音频表示单元的显示时间时刻可能有交叠,该交叠小于 0 到 2 个音频帧(见图 97 中所示的“音频交叠”)。将选择音频采样的指示,并且音频表示单元的显示与结合点后面的校正时间基准的再同步在播放器上设置。

经由所说明的 DVR-STD 的系统时间时钟，在时间 T5 显示 TS1 的最后音频表示单元。系统时间时钟可以在时间 T2 和时间 T5 之间交叠。在这个时间域期间，DVR-STD 在旧时间基准 (STC1) 的值和新时间基准 (STC2) 的值之间切换系统时间时钟。STC2 的值可以根据下列公式计算：STC2=STC1-

5 STC\_delta。

现在说明缓冲连续性。当第一个视频数据包的第一个字节到达 DVR-STD 的 TB1 时，STC11video\_end 是系统时间基准 STC2 上的 STC 的值。当第一个视频数据包的第一个字节到达 DVR-STD 的 TB1 时，STC22video\_start 是系统时间基准 STC2 上的 STC 的值。STC21video\_end 是按照系统时间基准 STC2 的 STC2 上的值计算的 STC11video\_end 的值。STC21video\_end 按照下列方程计算：STC21video\_end= STC11video\_end- STC\_delta。

10 为了遵从 DVR-STD，下列两个条件必须满足：第一，在 TB1 处的 TS2 的第一个视频数据包的到达时刻必须满足下列不等式：

$$\text{STC22video\_start} > \text{STC21video\_end} + \Delta T1。$$

15 如果有必要以满足上述不等式的方式再编码和/或多路复用 Clip1 和/或 Clip2 的部分流，则这个再编码或多路复用可以适当地得到执行。

第二，由从系统时间基准的时间轴上的 TS2 的视频数据包的输入跟随的、从相同时间轴上的 STC1 和 STC2 映射的、从 TS1 的视频数据包的输入，不必溢出或下溢视频缓冲器。

20 如果上面的句法、数据结构和规则被用作基础，则可以适当地管理记录于记录介质上的数据内容或再现信息，使用户能够在再现时确认记录于记录介质上的数据内容，或相当容易地再现期望的数据。

接下来将给出在图 46 中所示的 ClipInfo 的句法结构中的 time\_controlled\_flag (时间控制标志) 被设置为 1 的情况下，对记录 AV 25 流文件的详细描述。在 time\_controlled\_flag 被设置为 1 的情况下，AV 流的时间流逝和 AV 流的数据字节量具有下列的关系。即，假设在预设的误差范围内，时间流逝将正比于 AV 流的数据字节量。

$$\text{TS\_average\_rate} * 192 / 188 * (t - \alpha) \leq \text{AV\_file\_size}(t)$$

$$\leq \text{TS\_average\_rate} * 192 / 188 * (t + \alpha)$$

30

方程 (1)

尽管形式上有些较小的差别，上面的方程与图 46 的 ClipInfo 的

time-controlled-flag 的描述中所示的方程基本上相同。

在该方程中,TS-average-rate 是以字节/秒为单位所示的 AV 流文件(DVR 传输流文件)的平均位速率,并由 ClipInfo 中的相同名称的字段指示。另外,  $t$  是从以秒为单位所示的 AV 流文件的第一个源数据包开始的到达时间基准的流逝时间。AV\_file-size( $t$ )是以字节为单位所示的在时间  $t$  的 AV 流文件的大小。 $\alpha$ 是预定的固定值,例如 300 秒。

TS-average-rate 的值根据记录器的应用任意确定。例如,根据诸如长时间画面记录模式(LP 模式)、标准画面记录模式(SP 模式)、和高质量画面记录模式(HQ 模式)的记录模式为各自的模式确定 TS-average-rate 的值。

在记录 AV 流文件以满足方程(1)的情况下,如果该流被部分地擦除一定时间的流,可以确保所述的流的 TS-average-rate 指示的位速率在盘上产生可记录的时间等于所擦除时间的未占用区域。例如,在标准模式中,如果该流被部分地擦除一定时间的 AV 流,在盘上产生可记录时间等于相同标准模式中的擦除时间的未占用区域。

图 98 是用于说明在控制可变位速率以便时间流逝在预设误差范围内与 AV 流的数据字节数量成正比的情况下,图 1 的记录和/或再现设备的 AV 编码器 15 的操作的方块图。图 98 中用相同标号表示的方框与图 1 中的相同。

首先,诸如 LP 或 SP 模式的记录模式经由用户接口 24 输入至控制器 23。控制器 23 响应于该记录模式以设置视频编码的平均位速率和要记录的 AV 流(DVR 传输流)的多路复用位速率(图 99 的流程图的步骤 20)。

控制器 23 设置 time-controlled-flag 为 1,以将多路复用流的多路复用位速率和平均位速率分别设置为 TS-recording-rate 和 TS-average-rate。控制器 23 输出 ClipInfo 中的 Clip 信息文件的数据库,该 ClipInfo 的 time-controlled-flag、TS-recording-rate 和 TS-average-rate 被设置。Clip 信息文件经由图 1 中所说明的 ECC 编码单元 20 的处理被记录。

在编码模拟视频输入中,从终端 11 输入视频。在数字广播的视频输入代码转换中,从 AV 解码器 27 输入视频。输入视频被输入到视频编码器 151。控制器 23 计算分配给每个预设时间的视频的编码位的数量,以指定给视频编码器。视频编码器 115 编码每个预设时间的视频以将实际产生的位的数量输入给控制器 23。预设时间大小是视频 GOP 并等于 0.5 秒。根据从编码自

编码器输入的实际产生的编码位的数量的开始的累积值，控制器 23 控制视频编码的可变位速率，使得 AV 流的时间流逝将在预设误差范围内与 AV 流的数据字节数量成正比，以为下个预设时间的视频计算编码的位数量。如果控制器 23 可以从编码器输入相对视频编码难度（诸如运动矢量的预测剩余的  
5 量级或 DCT 系数的量化比例的量级）而形成，则可以实现可变位速率以得到更高的画面质量。即进行控制使得相对视频编码难度越高，分配给每个预设时间的视频的编码位的数量越大。

视频编码器 115 将视频流输入给多路复用器 16，该多路复用器还被馈送有音频流和诸如 AV 同步的系统信息（S）。音频输入或诸如 AV 同步的系统  
10 信息的编码处理流程与图 1 的说明相同。

多路复用器 16 多路复用视频和音频流以传输预设多路复用位速率的流。在将视频和音频分包的过程中，必须进行控制以便不破坏 MPEG2 传输流的系统目标解码器（T-STD）。如果由于 T-STD 限制，不可能将视频访问单元（编码的 I、P 和 B 画面的画面）、也不可能将音频访问单元（音频帧）分包，  
15 则多路复用器 16 多路复用以不产生空数据包（为 0x1FFF 的数据包）。通过这个多路复用控制，相邻传输数据包之间的时间间隔变得不规则，使得数据包间断地产生。

从多路复用器 16 输出的传输数据包被输入到源分包器 19，然后源分包器 19 将到达时间戳附加到每个传输包以形成源数据包。向前填充（front-  
20 padded）源数据包串以产生 AV 流文件，该流文件然后经由 ECC 编码单元 20 中的处理被记录在记录介质上。

图 99 是用于说明在保证 AV 流的时间流逝将在预设的误差范围内与 AV 流的数据字节的数量成正比的编码模式（time-controlled-flag=1）中，以编码视频的可变位速率记录 AV 流的操作的流程图。

25 在步骤 20，控制器 23 设置视频编码的平均位速率和传输流的多路复用位速率 TS-average-rate。

视频编码的平均位速率是 TS-average-rate，小于音频编码的常数位速率，小于多路复用开销位速率。注意，根据记录器应用程序（LP 或 SP 模式）将 TS-average-rate 设置为预定值。

30 TS-recording-rate 大于以可变位速率的视频编码的最大位速率加上音频编码的常数位速率加上多路复用开销位速率。

在步骤 21, 控制器 23 控制视频编码器 151, 使得视频流以可变位速率按如下方式被编码, 即从一个预设时间域到另一个预设时间域保证预设的平均位速率。

在步骤 22, 控制器 23 控制多路复用器 16, 以便在基本流不生成传输数据包的情况下, 不产生空数据包。通过这个多路复用控制, 相邻传输数据包之间的时间间隔变得不规则, 使得数据包间断地产生。

在步骤 23, 控制器 23 将到达时间戳附加到每个传输包以通过控制源分包器 19 形成源数据包。向前填充源数据包串以产生 AV 流文件, 该流文件然后被记录。

10 下面说明在可变位速率编码视频的情况下, 控制 MPEG 的 VBV (视频缓冲校验器) 的方法。注意, VBV 是 MPEG (见图 100) 规定的理论上的解码器模型。是 MPEG 解码器解码视频流使得 VBV 将正确地操作。这限制了编码方法(主要是量化控制和对画面的位数量的限制)。VBV 所拥有的缓冲器称为 VBV 缓冲器。这代表真实解码器的理论上必要的最小的缓冲器大小。在 MPEG2 主  
15 简档 (profile) 主级别的情况下, VBV 缓冲器大小是 1.75Mbits (兆位)。

至于对 MPEG 的 VBV 的可变位速率, 图 101 中所示的方法是广泛公知的。即, 图 101 说明了 VBV 控制, 其情况是: 如果 VBV 缓冲器中的可变位速率不满, 到缓冲器的输入位速率变为最大, 如果 VBV 缓冲器中的可变位速率为满, 到缓冲器的输入位速率变为零。在图 101 中, 右侧上升线的倾斜度代表 VBV  
20 的最大位速率。如果 VBV 缓冲器未滿, 缓冲器占用 (takeup) 量以 VBV 的最大位速率上升。另一方面, 如果 VBV 缓冲器中的位占用 (occupying) 量已滿, 则到缓冲器的输入位速率为 0, 缓冲器占用量保持不变。横坐标表示时间轴, T1 表示解码时间点。在时间 T1, 时间点 T1 的画面被立即解码, 而缓冲器占用量减少。接着, 画面在每个预设时间间隔被解码以减少缓冲器占用  
25 量。在图 101 中所示的方法中, 没有视频解码器在视频流中填充字节的可能性。

另一方面, 根据本发明, 如图 102 中所示控制 VBV。即, 在以可变位速率改变每个预设时间间隔的位速率的过程中, 在预设时间内以 CBR (常数位速率) 执行 VBV 控制。图 102 显示了在 GOP (诸如 0.5 秒视频序列) 内的 CBR  
30 控制的情况下的 VBV 控制。即, 图 102 说明了在到 VBV 缓冲器的输入位速率是当前 GOP 的编码位速率的情况下和在插入填充字节以便不产生 VBV 缓冲器

的溢出的情况下的 VBV 控制。

下面的序列用于校验是否插入填充字节和在插入填充字节的情况下计算填充字节的数量。在下列说明中, 假设

VBV-BUFFER-SIZE=1.75\*1024\*1024 位

5      gop-bit-rate: 每个 GOP 的位速率 (位/秒)。

(1) 正在被编码的画面的最小位数量的计算

取图 102 的时间 d1 处的画面为例。首先, 获得紧接在通过时间 t1 处的画面的 VBV 解码之前的 VBV 缓冲器的位占用量 vbv-b。然后计算对应于位占用量 vbv-d 加上以位速率 gop-bit-rate 从时间 d1 到下一个画面的解码时间  
10      d2 (tau) 输入的位数量的和数 tmp。当前被编码的画面的最小位数量可以从 tmp 和 VBV-BUFFER-SIZE 得出如下:

tmp=vbv-b+gop-bit-rate\*tau

min-picture-bit=tmp- VBV-BUFFER-SIZE

(2) 检查接着画面编码是否需要字节填充

15      如果当前画面的实际编码位 gen-picture-bit 小于 min-picture-bit, 则产生由下列公式指示大小的填充字节。在画面被当前编码后, 视频编码器编码多个等于 num-stuffing-byte 的数的填充字节。一个填充字节是 8 位代码 “0000 0000”。

If (如果) (gen-picture-bit<min-picture-bit)

20      num-stuffing-byte=( min-picture-bit-gen-picture-bit+4 ) /8。

在图 102 中所示的方法中, 目的是进行控制使得视频编码器使用分配给预设时间的视频的位数量。到 VBV 缓冲器的输入位速率是当前 GOP 的编码位速率, 然后视频编码器产生填充字节以便不在 VBV 缓冲器中产生溢出。

根据本发明的概念, 如图 103 所示, VBV 控制器在这个目的上是有效的,  
25      即 AV 流的时间流逝和 AV 流的字节数量之间的关系将在预设的误差范围内呈线性。如果使用图 101 中所示的 VBV 控制, 则在输入视频中持续很长时间的静态画面的显示中不能保证图 103 中所示的关系。即, 静态画面具有较少数量的信息, 使得如果将用于编码所分配的位数量做得大于信息的数量, 则将使实际编码中产生的位数量以较小的值饱和。因此, 在这种情况下, AV 流  
30      的时间流逝和 AV 流的数据字节量之间的关系不是线性的, 如图 104 所示。如果在这种情况下, 使用图 102 中所示的 VBV 控制, 则 AV 流的时间流逝和

AV 流的数据字节量之间的关系在预设的误差范围内可以肯定是线性的，因为目标在于管理控制以使用分配给预设时间的视频的位数量的视频编码器产生填充字节，使得到 VBV 缓冲器的输入位速率将是当前 GOP 的编码速率并使得将不在 VBV 缓冲器中出现溢出。

- 5 在图 104 的情况下，如果擦除静态画面的部分时间的 AV 流，则不可能在盘上产生可记录对应于具有的 TS-average-rate 指示的位速率的已擦除的部分时间的未占用区域，因为该部分时间的数据字节数量比对应于平均位速率乘以擦除时间的数据字节小。另一方面，如果该流被部分地擦除该 AV 流的一定时间，则在盘上可能产生可记录等于以由所述的流的 TS-average-rate 指示的位速率的擦除时间的未占用区域。

图 105 显示了用于说明在上述图 99 的步骤 S21 的处理中的视频的可变位速率控制的详细处理的流程图。

- 15 在步骤 S200，在 VBV 的公差值 sv-now 中设置初始值 SV1。根据本发明，为了保证 AV 流的时间流逝和 AV 流的数据字节数量之间的关系在预设误差范围内将呈线性，管理可变位速率控制，使得 VBV 的公差值 SV-now 将处于 0 和最大值 SVMAX 的范围内。

如果在上面的方程 (1) 中， $\alpha=300$  秒，SV1 和 SVMAX 是下面的值：

$$SV1 = (\text{视频的平均位速率}) * 300$$

$$SVMAX = SV1 * 2$$

- 20 这里视频的平均位速率是在图 99 的步骤 S20 确定的值 (见图 107)。

在步骤 S201，计算当前 GOP 的编码所分配的位数量。

在步骤 S202，检查下列不等式是否成立：

$$sv\_now + b\_av - b\_alloc \leq 0.$$

执行这个检查以校验 VBR 的公差值是否为负 (-)。

- 25 在上面的不等式中，b-av 是从视频的平均编码位速率得出的为每个 GOP 所分配的编码位的数量的平均值。如果 GOP 的时间期间是 0.5 秒，b-av 是下面的值：

$$b\_av = (\text{视频的平均位速率}) * 0.5$$

- 30 如果步骤 S202 的检查结果是 YES (是)，处理转移到步骤 S203。如果步骤 S202 的检查结果是 NO (否)，则处理转移到步骤 S204，在那里将 b-alloc 的值设置为 b-av。然后处理转移到步骤 S205。



在步骤 S203, 检查下列不等式是否成立:

$$sv\_now + b\_av - b\_alloc \leq SVMAX$$

如果步骤 S203 的检查结果是 YES (是), 处理转移到步骤 S205。如果步骤 S203 的检查结果是 NO (否), 则处理转移到步骤 S204, 以将  $b\_alloc$  的值设置为  $b\_av$ 。然后处理转移到步骤 S205。

在步骤 S205, 编码当前的 GOP。管理控制使得用分配的位数量  $b\_alloc$  编码当前的 GOP 和使得将不在 VBV 缓冲器中产生溢出。将参照图 106 详细说明这个处理。

在步骤 S206, 根据下列方程更新 VBR 的公差值  $sv\_now$ :

$$sv\_now += b\_av - b\_gen$$

这里  $b\_gen$  是作为编码当前 GOP 的结果而获得的当前 GOP 的编码位的数量。

在步骤 S207, 检查当前 GOP 是否是最后的 GOP。如果步骤 S207 的检查结果为是, 处理结束。如果步骤 S207 的检查结果是否, 则处理返回到步骤 S201。

图 106 是用于说明在图 105 的步骤 S205 的处理中的 VBV 控制的详细处理的流程图。

在步骤 S300, 根据下列方程将分配给当前 GOP 的编码位的数量转换为编码位速率  $gop\_bit\_rate$ :

$$gop\_bit\_rate = b\_alloc / (15 / 29.97)。$$

在步骤 S301, 正在当前 GOP 中编码的画面的最小位数量  $min\_picture\_bit$  通过下列方程得出:

$$tmp = vbv\_b + gop\_bit\_rate * \tau$$

$$min\_picture\_bit = tmp - VBV\_BUFFER\_SIZE。$$

在上面的方程中,  $vbv\_b$  是紧接在当前由 VBV 编码的画面的解码的前面的 VBV 缓冲器的位占用数量 (见图 102)。

注意,  $\tau$  是当前被编码的画面的解码时间和下一个画面的解码时间之间的差 (见图 102)。

注意,  $VBV\_BUFFER\_SIZE$  是 VBV 缓冲器的大小, 并且在 MPEG2 MP@ML 的情况下是 1.75Mbits。

在步骤 S302, 编码当前的画面以产生生成的位数量  $gen\_picture\_bit$ 。

在步骤 S303, 检查下列不等式:

$\text{gen-picture-bit} < \text{min-picture-bit}$

如果步骤 S303 的检查结果是 YES (是), 处理转移到步骤 S304。如果步骤 S303 的检查结果是 NO (否), 则处理转移到步骤 S305。

- 5       在步骤 S304, 视频编码器在当前编码的画面之后编码多个对应于  $\text{num-stuffing-byte}$  的数量的填充字节, 并将编码的填充字节附加在编码的画面的后部 (见图 102):

$$\text{num-stuffing-byte} = (\text{min-picture-bit} - \text{gen-picture-bit} + 4) / 8$$

- 10       在步骤 S305, 检查画面是否是 GOP 的最后的画面。如果步骤 S305 的检查结果是 YES (是), 处理结束。如果步骤 S305 的检查结果是 NO (否), 则处理返回到步骤 S301。

- 15       通过如上所述控制视频流的可变位速率编码来生成 AV 流文件, 可以保证 AV 流的时间流逝和 AV 流的数据字节数量之间的关系在预设的误差范围内为线性。如果在这种方式中, 部分地擦除了流的一定时间部分, 则可能产生在数量上对应于所擦除的时间区域的未占用区域, 该未占用区域可以用由所述的流的  $\text{TS-average-rate}$  指示的位速率可记录。

为了比较, 现在显示不保证 AV 流的时间流逝和 AV 流的数据字节数量之间的正比关系 ( $\text{time-controlled-flag}=0$ ) 的编码模式中的 AV 流的两个示例性记录方法。

- 20        $\text{time-controlled-flag}=0$  的第一种情况是透明的记录数字广播的 AV 流 (节目) 的传输流的情况。如果数字广播使用统计多路复用, 那里的 AV 流常常是可变位速率。AV 流的时间流逝和 AV 流的数据字节数量之间的关系通常不必是线性的, 使得如果在这个 AV 流的透明的记录上制备 Clip, 则将  $\text{time-controlled-flag}$  设置为 0。

- 25        $\text{time-controlled-flag}=0$  的第二种情况是以可变位速率编码视频使得该位速率将不高于预设平均位速率的情况。这种情况是, 在其中执行视频编码的 VBV 控制, 使得如果 VBV 缓冲器不满或已满, 到缓冲器的输入位速率将分别是最大位速率或零。现在参照图 108 至 109 说明这种情况中的 AV 流记录方法。

- 30       图 108 显示了用于说明在不保证 AV 流的时间流逝和 AV 流的数据字节数量之间的正比关系的编码模式中以可变位速率编码视频以记录 AV 流的操作

的流程图。

除步骤 S400 外，该流程图与图 99 的流程图相同。

在步骤 S400，控制视频编码器 151 使得视频流将以可变位速率用如下方式进行编码，该方式为该位速率将低于从一个预设时间域到另一个预设时间域的预设平均位速率。

图 109 是用于说明图在图 108 的步骤 S400 的处理中的视频的可变位速率控制的详细处理的流程图。

在步骤 S500，在 VBR 的公差值 `sv-now` 中设置初始值 `SV1`。在这种情况下进行可变位速率控制使得公差值 `sv-now` 将不为负 (-)。

10 在步骤 S501，得出当前 GOP 的编码所分配的位 `b-alloc`。

在步骤 S502，检查下列不等式是否成立：

$$sv\_now + b\_av - b\_alloc \geq 0$$

这个步骤 S502 是用于校验 VBV 的公差值是否不为负 (-) 的检查步骤。

在上面的不等式中，`b-av` 是编码每个 GOP 的编码位的数量的平均值。

15 如果所述的 GOP 的时间期间是 0.5 秒，`b-av` 具有下面的值：

$$b\_av = (\text{视频的平均位速率}) * 0.5$$

如果步骤 S502 的检查结果是 YES (是)，处理转移到步骤 S504。如果步骤 S502 的检查结果是 NO (否)，则处理转移到步骤 S503，以将 `b-alloc` 设置为 `b-av`。然后处理转移到步骤 S504。

20 在步骤 S504，编码当前的 GOP。用分配的位数量 `b-alloc` 编码当前的 GOP。这种情况下的 VBV 管理控制是这样的，如果 VBV 缓冲器中的位占用值不满或已满，到缓冲器的输入位速率将分别是 VBR 的最大位速率 (可变位速率) 或零 (见图 101)。在这个步骤中，不在视频流中编码填充字节。

在步骤 S505，根据下列方程更新 VBR 的公差值 `sv-now`：

25 
$$sv\_now += b\_av - b\_gen$$

这里 `b-gen` 是作为在步骤 S504 编码当前 GOP 的结果而获得的当前 GOP 的编码位的数量。

在步骤 S506，检查当前 GOP 是否是最后的 GOP。如果步骤 S506 的检查结果为是，则处理结束。如果步骤 S506 的检查结果是否，则处理返回到步骤 S501。

30 图 108 和 109 的记录方法不能保证 AV 流的时间流逝和 AV 流的数据字节

数量之间在预设误差范围内的正比关系。例如，如果有一个静态画面在输入视频中持续很长时间，则 AV 流的时间流逝和 AV 流的数据字节数量之间的关系如图 104 所示。即，由于静态画面具有较小的信息量，即使用于编码所分配的位数量大于该信息量，实际编码中产生的位数量也以较小的值饱和。因此，在这种情况下，AV 流的时间流逝和 AV 流的数据字节数量之间的关系不是线性。

另一方面，如果考虑到管理控制使得视频编码器将使用预设时间分配的位数量，执行控制使得得到 VBV 缓冲器的输入位速率是当前 GOP 的编码位速率，并使得将由视频编码器产生填充字节以便不产生 VBV 缓冲器溢出，可以保证 AV 流的时间流逝和 AV 流的数据字节数量之间的关系在预设误差范围内基本上为线性。

作为实现保证 AV 流的时间流逝和 AV 流的数据字节数量之间的正比关系的编码模式的方法，可以考虑在多路复用传输数据包中插入空数据包，以记录常数位速率的传输流。这是主要用于记录诸如 D-VHS 的传输数据包的编码方法。同时，空数据包意味着其数据包 ID (PID) 被设置为 0x1FFF 并且不具有信息意义的传输数据包。

为比较该方法与图 99 中的方法起见，图 110 显示了编码模式的流程图，其中通过编码预设位速率的传输流，保证了 AV 流的时间流逝和 AV 流的数据字节数量之间的正比关系。

在步骤 S600，设置传输流的多路复用位速率和视频编码的位速率。在步骤 S601，以预设常数位速率或以低于预设位速率的位速率编码视频流。

在步骤 S602，在缺少用于生成传输数据包的基本流的情况下，生成并多路复用空数据包（没有信息意义的传输数据包），以编码预设常数多路复用位速率的传输流。

在步骤 S603，将到达时间戳附加于每个传输数据包，以形成源数据包，然后将该源数据包记录在记录介质上。

如果通过上述记录方法将 AV 流记录为 Clip，将 Clip 的 time-controlled-flag 设置为 1。可是，使用空数据包并且因此不能使用用于有效视频编码的编码位的这种方法招致了画面质量低于用图 99 的编码方法所取得的画面质量的问题（这已经在例如 Japanese Laying-Open Patent Publication（日本待审专利）H-11-220727 号的现有技术的说明中进行了讨论）。因此，本

发明中不推荐使用图 111 的记录方法。

下面说明仅部分删除 AV 流文件的一定时间部分的方法。

图 111 显示了原始 AV 流文件和用于擦除 AV 流文件的部分再现部分接续编辑的 AV 流文件的示例。假设, 在编辑之前, Virtual PlayList (虚拟播放列表) 指向原始 AV 流文件上的 IN-time 和 OUT-time。如果执行未被 Virtual PlayList 使用的擦除流部分的编辑 (最小化编辑), 它将原始 AV 流文件改变为图 111 中所示的经编辑 (as-edited) 的流文件。擦除了从原始 AV 流文件的前端到点 X 的数据和从点 Y 到尾端的数据。下面说明确定这些点 X 和 Y 的一般方法。

10 图 112 说明了不用分析 AV 流内容而擦除点 IN 之前的无用数据的方法。PlayList 表示原始 AV 流文件上的点 IN, 并表示 AV 流的 EP-map。为了解码由点 IN 表示的画面, 需要从地址 ISA2 开始的 I 画面。

在点 X 的后部, 需要 PAT、PMT 和 PCR 数据包。RSPN-EP-start=ISA1 的 PTS 是 pts1, 而 RSPN-EP-start=ISA2 的 PTS 是 pts2。如果 pts1 的系统时间基准和 pts2 的系统时间基准的时间差不小于 100 毫秒, 则在地址 ISA1 和 ISA2 之间存在 PAT、PMT 和 PCR (这至少对 SESF、DVS、ATSC、和 ISDB 有效)。

因此, 在地址 ISA1 的前面确定点 X。点 X 必须在联合单元的边界上。使用 EP-map, 而不用分析 AV 流的内容, 记录器通过下列步骤也能确定点 X:

(S1) 发现具有与系统时间基准上的 IN-time 的 PTS 最近的过去显示时间的 PTS 值的 SPN-EP-start 的步骤; 和

(S2) 发现具有比在步骤 S1 发现的 SPN-EP-start 的 PTS 值再过去至少 100 毫秒的显示时间的 PTS 值的 SPN-EP-start 的步骤;

(S3) 在步骤 S2 发现的 SPN-EP-start 的前面确定点 X, 点 X 必要地位于联合单元的边界上。

25 这种方法简单, 因为不需要读出 AV 流的数据以分析它的内容。可是, 在再现所编辑的 AV 流的 PlayList 时, 有遗留无用数据的情况。如果为了确定点 X 读出 AV 流数据并分析数据内容, 则用于再现 PlayList 的无用数据可以被有效地擦除。

图 113 说明了不用分析 AV 流内容而在 OUT 点的后部擦除无用数据的方法。PlayList 指向原始 AV 流上的 OUT 点, 并显示 AV 流的 EP-map。

预先假定从 SPN-EP-start=ISA4 开始的视频序列是:

I2 B0 B1 P5

这里 I、P 和 B 分别表示 I-、P-和 B-画面。并且后缀号码表示显示次序。如果在这个处理中，记录器不能分析 AV 流内容，则它就不知道诸如画面编码类型或参照 OUT-time 的 PTS 的画面的时间基准的信息。OUT-time 的 PTS 可能正在由画面 B0 或 B1 参照（这不能被知道，除非记录器分析 AV 流内容）。在这种情况下，I2 是必要的，以便解码画面 B0 和 B1。I2 的 PTS 大于 OUT-time 的 PTS。I2 的 PTS 大于 OUT-time 的 PTS ( $\text{OUT-time} < \text{pts4}$ ，这里 pts4 是 I2 的 PTS)。尽管 I2 的 PTS 大于 OUT-time 的 PTS，I2 对于 B0 和 B1 是必要的。

因此，确定点 Y 位于所示的地址 ISA5 的后部。ISA5 是直接位于 EP-map 中的 ISA4 后部的 SPN-EP-start 的值。点 Y 也必须位于联合单元的边界上。

使用 EP-map，而不用分析 AV 流的内容，记录器通过下列步骤可以确定点 Y:

(S1) 发现具有与系统时间基准上的 OUT-time 的 PTS 最近并且比其更超前的显示时间的 PTS 值的 SPN-EP-start 的步骤；和

(S2) 发现具有直接位于在系统时间基准上发现的 SPN-EP-start 后部的 SPN-EP-start 的步骤；

(S3) 设置点 Y 以便其比在步骤 S2 发现的 SPN-EP-start 更靠后。点 Y 必须位于联合单元的边界上。

这种方法简单，因为不需要读出 AV 流的数据以分析它的内容来确定点 Y。可是，在再现所编辑的 AV 流的 PlayList 时，有遗留无用数据的情况。如果为了确定点 Y 读出 AV 流数据并分析数据内容，用于再现 PlayList 的无用数据可以被更加有效地擦除。

参照图 114 的流程图说明 PlayList 的一般操作。在图 1 所示的记录和/或再现设备的多路复用流分析单元 18 上执行这个处理。

在步骤 S11，流分析单元 18 设置记录的 AV 节目的视频的 PID。如果传输流中包括多个视频流，则设置各自的视频 PID。

在步骤 S12，流分析单元 18 接收视频传输数据包。

在步骤 S13，流分析单元分析传输数据包的有效载荷（紧接数据包标头的的数据部分）是否以 PES 数据包（PES 数据包是 MPEG2 中规定的数据包并分包基本流）的第一个字节开始。这可以通过检查传输数据包标头中的“payload-unit-start-indicator(有效载荷开始指示符)”而得知。如果这

个值是 1, 传输数据包的有效载荷以 PES 数据包的第一个字节开始。如果步骤 S13 的检查结果为否, 则处理返回到步骤 S12。如果步骤 S13 的检查结果为是, 则处理转换到步骤 S14。

在步骤 S14, 流分析单元检查 PES 数据包的有效载荷是否开始于 MPEG 视频的 sequence-header-code(序列标头编码)的第一个字节(32 位的“0x000001B3”)。如果步骤 S14 的检查结果为否, 则处理返回到步骤 S12。如果步骤 S13 的检查结果为是, 则处理前进到步骤 S15。

在步骤 S15, 当前传输数据包是入口点。在步骤 S16, 流分析单元获得开始于 sequence-header-code 的 I 画面的 PTS 所属的视频的 PID 及其入口点以及上述数据包的数据包号码, 以将视频 PID 输入控制器 23。控制器 23 形成 EP-map。

在步骤 S17, 检查当前数据包是否是最后输入的传输数据包。如果当前数据包不是最后的数据包, 处理回到步骤 S12。如果当前数据包是最后的数据包, 则处理结束。

上述操作序列不仅可以在硬件上、也可以在软件上执行。如果操作序列在软件上执行, 形成软件的程序或多个程序安装在计算机的专用硬件上, 使得各程序从诸如通用个人计算机的记录介质上进行安装。

参照图 115, 这个记录介质除了计算机外, 可以由向用户提供程序的包介质构成。包介质可以是例如磁盘 221(包括软盘)、光盘 222(包括 CD-ROM(致密盘-只读存储器)和 DVD(数字通用盘))、磁光盘 223(包括 MD(小型盘))或半导体存储器 224。另外, 记录介质可以由装备给用户的硬盘构成, 该硬盘预装载于计算机上并包括其中存储了程序的 ROM 202 或存储器 208。

在本申请文件中, 说明由介质提供的程序的序列的各个步骤不仅包括按照说明的顺序执行的处理, 也包括并行或成批执行的处理。

本申请文件中的系统表示由多个装置构成的整个设备。

## 工业应用性

在编码和记录 AV 流期间, time-controlled-flag 和 TS-average-rate 被记录为 AV 流的属性信息。如果将 time-controlled-flag 设置为 1, 则保证在 AV 流的时间流逝和 AV 流的数据字节数量之间的关系在预设误差范围内

是线性的。用字节/秒为单位的话，TS-average-rate 表示 AV 流文件（传输流）的平均位速率，并由记录器的应用程序在预设值处进行确定。例如，为各个模式设置 TS-average-rate，即长播放（LP）模式、标准播放（SP）模式、或高质量（HQ）模式。

- 5       如果将 AV 流文件的 time-controlled-flag 设置为 1，并部分地将流擦除一预设时间流部分，则例如可以在盘上创建对应于 SP 模式的 AV 流文件时间部分的未占用区域。

如果将 time-controlled-flag 设置为 1，则如下编码 AV 流：

- (1) 设置传输流的多路复用位速率和视频编码的平均位速率。
- 10   (2) 以可变位速率编码视频流，以便保证每个预设时间域的预设平均位速率。应该注意，MPEG 视频编码的 VBV（视频缓冲校验器）以如下方式受到控制，即为了管理控制使得视频编码器将使用分配给预设时间的视频的位数量，到 VBV 缓冲器的输入位速率将是当前编码位速率，并且使得视频编码器将生成限制 VBV 缓冲器的溢出的填充字节。
- 15   (3) 控制多路复用，使得在没有用于生成传输流的基本流的情况下，不生成空数据包。
- (4) 将到达时间戳附加到每个传输流以将它生成源数据包，并且向前填充源数据包，用于记录成 AV 流文件。

- 通过以这种方式编码和记录 AV 流文件，可以保证，如果部分地擦除了一定时间部分的流，则可以在盘上创建对应于所擦除时间部分的未占用区域，并且该未占用区域以由 TS-average-rate 所示的位速率可记录。
- 20



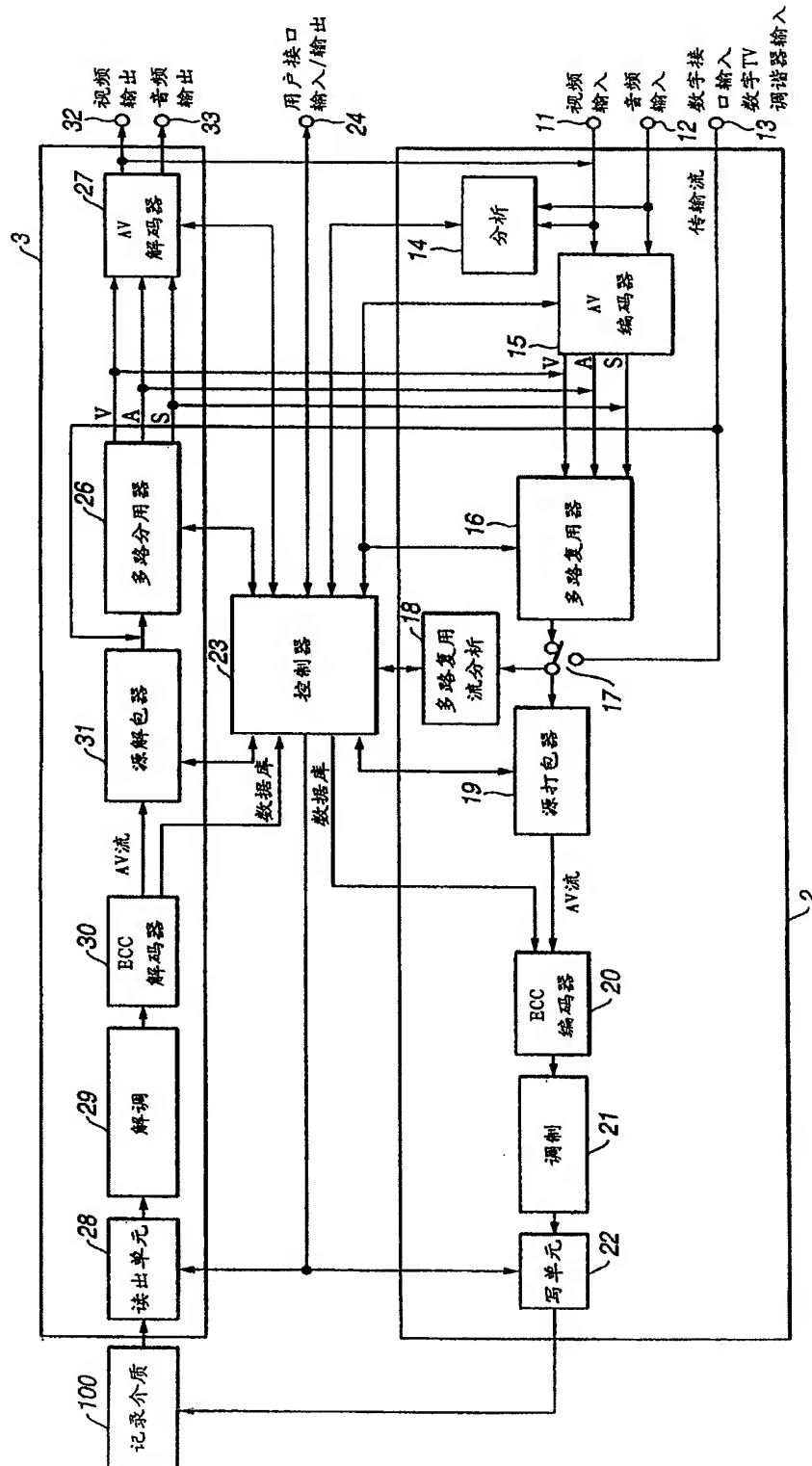


图 1

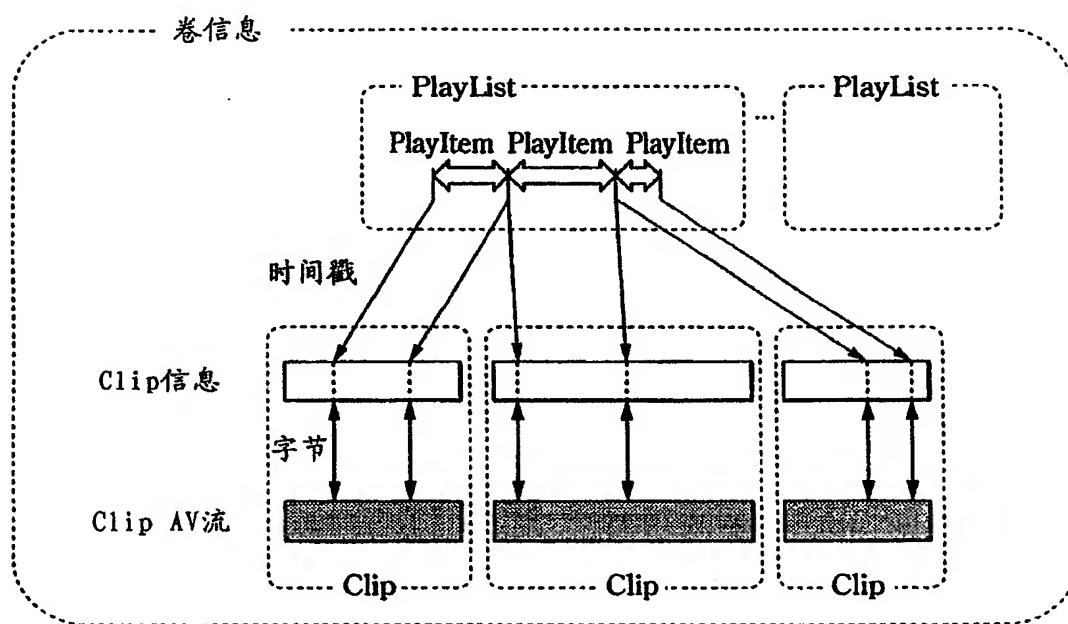


图 2

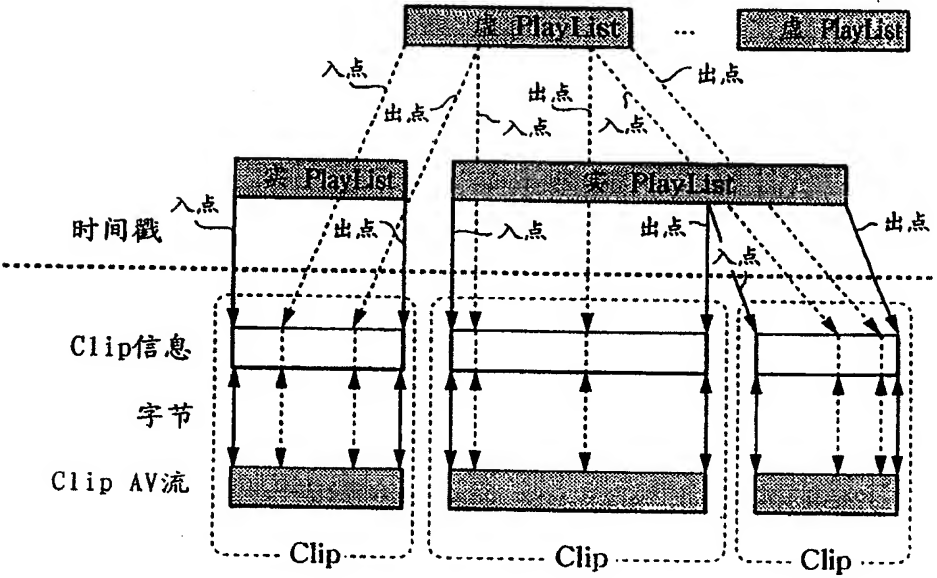


图 3

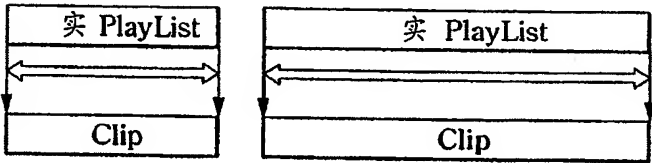


图 4A

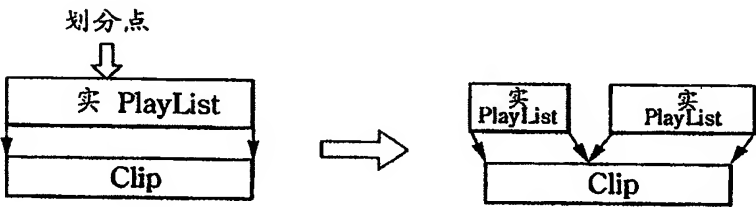


图 4B

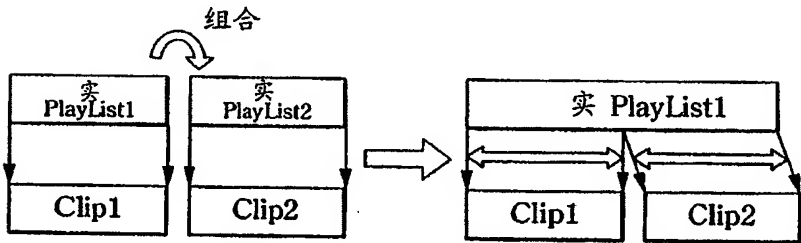


图 4C

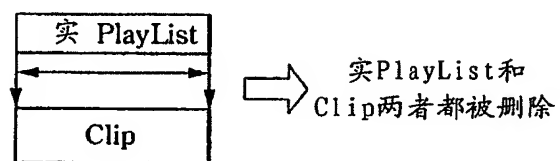


图 5A

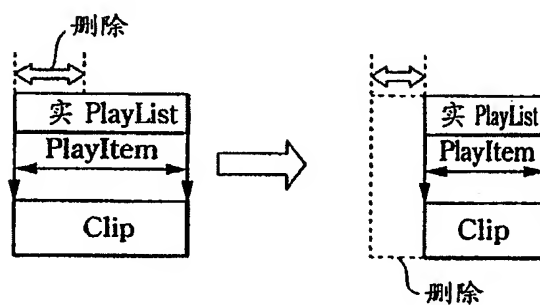


图 5B

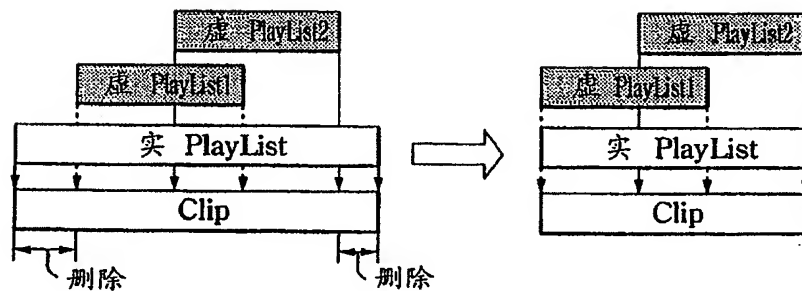
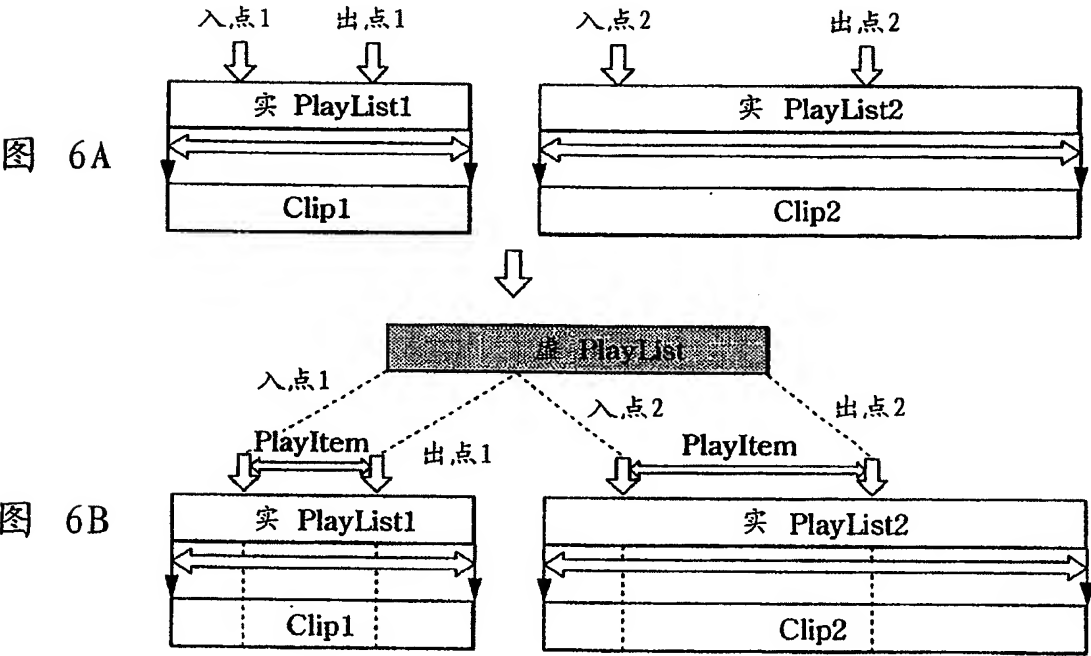


图 5C



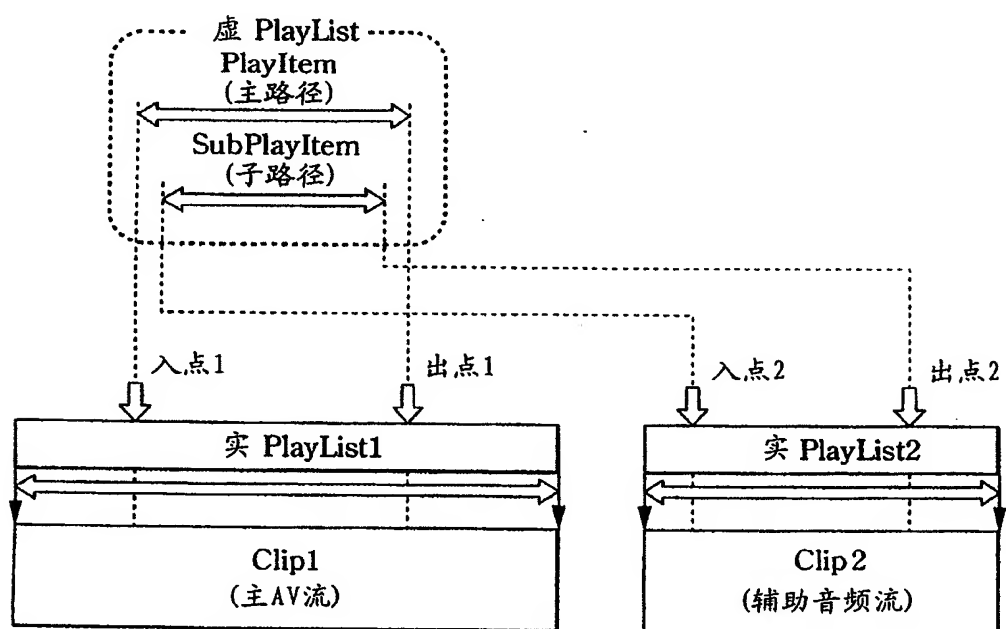


图 7

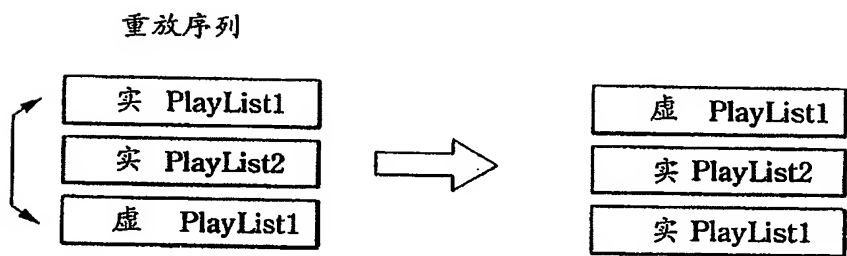


图 8



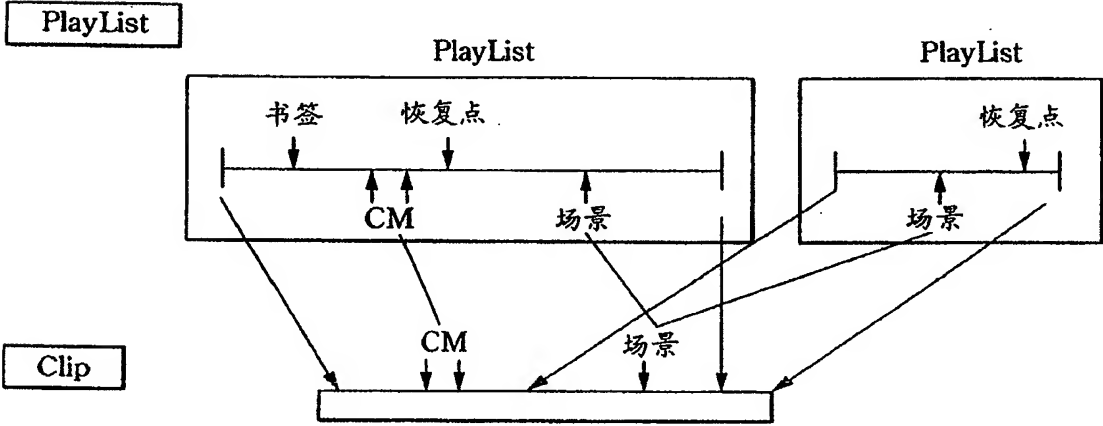
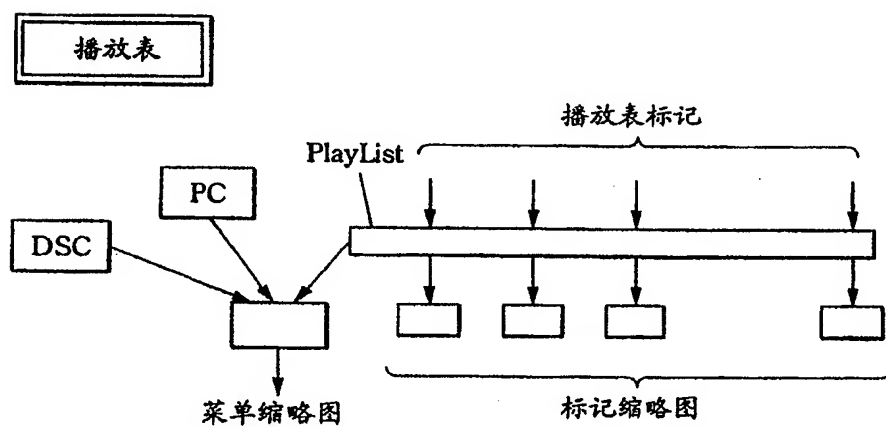
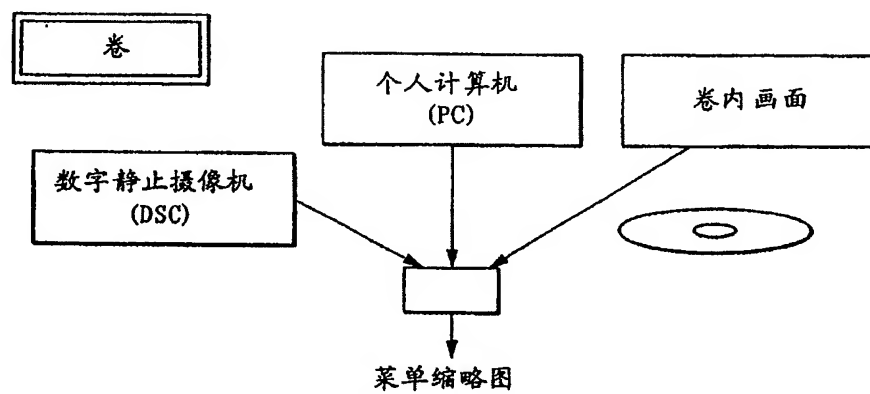


图 9



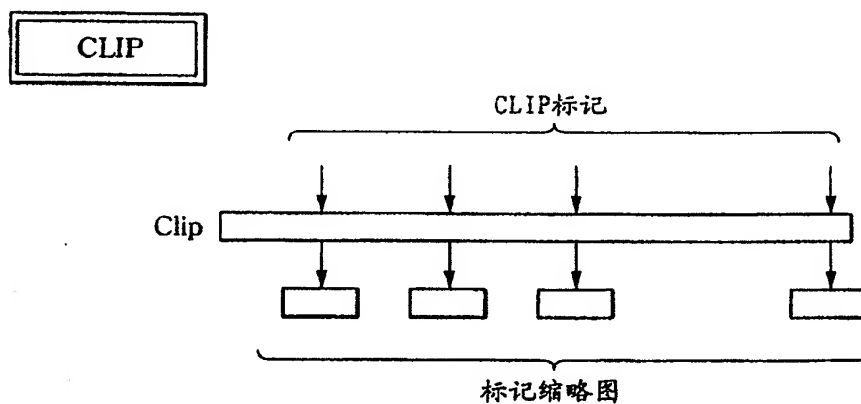


图 12

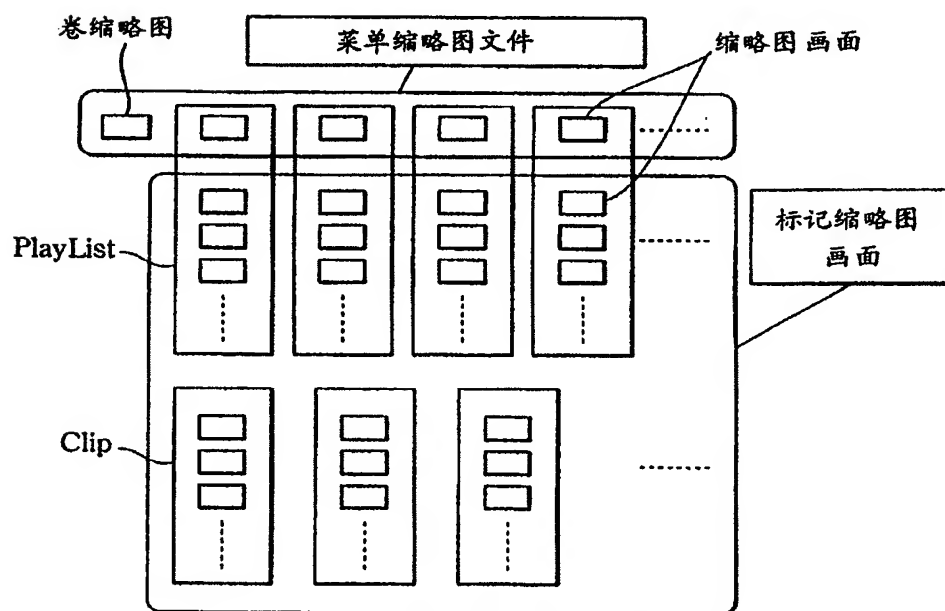


图 13

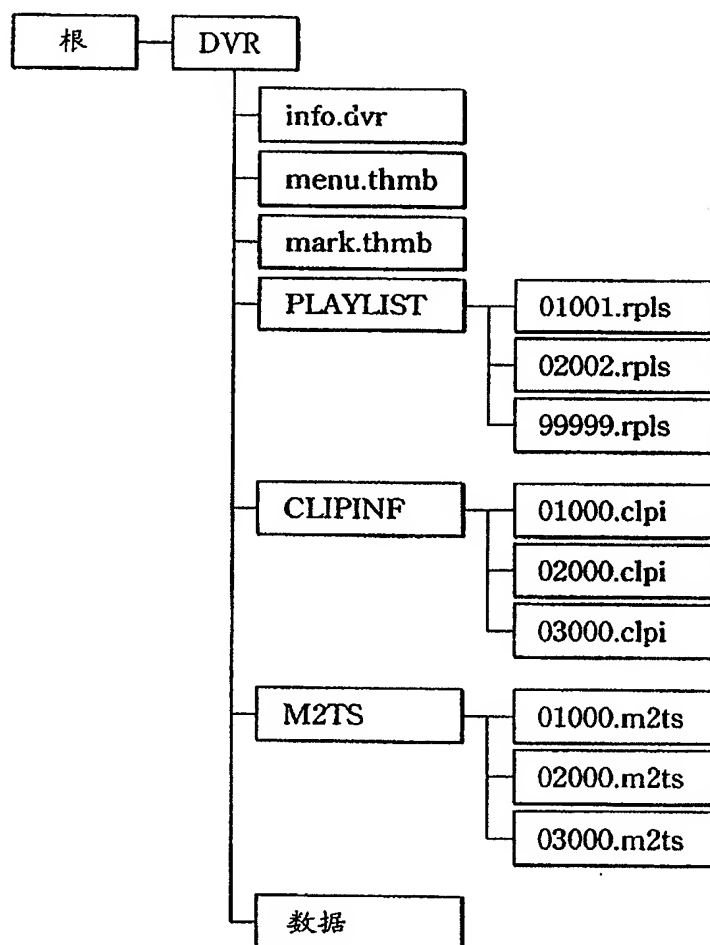


图 14

句法	字节数	缩写
info.dvr {		
<b>TableOfPlayLists_Start_address</b>	32	uimbsf
<b>MakersPrivateData_Start_address</b>	32	uimbsf
保留	192	bslbf
<b>DVRVolume()</b>		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
<b>TableOfPlayLists()</b>		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
<b>MakersPrivateData()</b>		
}		

图 15

句法	字节数	缩写
DVRVolume(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>ResumeVolume()</b>		
<b>UIAppInfoVolume()</b>		
}		

图 16

句法	字节数	缩写
ResumeVolume(){		
保留	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

图 17

句法	字节数	缩写
UIAppInfoVolume(){		
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>Volume_name</b>	8*256	bslbf
保留	15	bslbf
<b>Volume_protect_flag</b>	1	bslbf
<b>PIN</b>	8*4	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
<b>reserved_for_future_use</b>	256	bslbf
}		

图 18



值	字符字母编码
0x00	保留
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (单一码)
0x03-0xff	保留

图 19

句法	字节数	缩写
TableOfPlayLists(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayLists</b>	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
<b>PlayList_file_name</b>	8*10	bslbf
}		
}		

图 20

句法	字节数	缩写
TableOfPlayLists(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayLists</b>	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
<b>PlayList_file_name</b>	8*10	bslbf
<b>UIAppInfoPlayList()</b>		
}		
}		

图 21

句法	字节数	缩写
<b>MakersPrivateData(){</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>if (length !=0){</b>		
<b>mpd_blocks_start_address</b>	32	uimsbf
<b>number_of_maker_entries</b>	16	uimsbf
<b>mpd_block_size</b>	16	uimsbf
<b>number_of_mpd_blocks</b>	16	uimsbf
保留	16	bslbf
<b>for (i=0; i&lt;number_of_maker_entries; i++){</b>		
<b>maker_ID</b>	16	uimsbf
<b>maker_model_code</b>	16	uimsbf
<b>start_mpd_block_number</b>	16	uimsbf
保留	16	bslbf
<b>mpd_length</b>	32	uimsbf
<b>}</b>		
<b>stuffing_bytes</b>	8*2*L1	bslbf
<b>for (j=0; j&lt;number_of_mpd_blocks; j++){</b>		
<b>mpd_block</b>	mpd_block_ size*1024*8	
<b>}</b>		
<b>}</b>		

图 22

句法	字节数	缩写
xxxxx.rpls / yyyyy.vpls {		
<b>PlayListMark_Start_address</b>	32	uimsbf
<b>MakersPrivateData_Start_address</b>	32	uimsbf
保留	192	bslbf
<b>PlayList()</b>		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
<b>PlayListMark()</b>		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
<b>MakersPrivateData()</b>		
}		

图 23

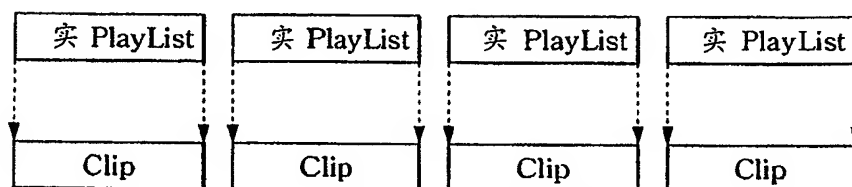


图 24A

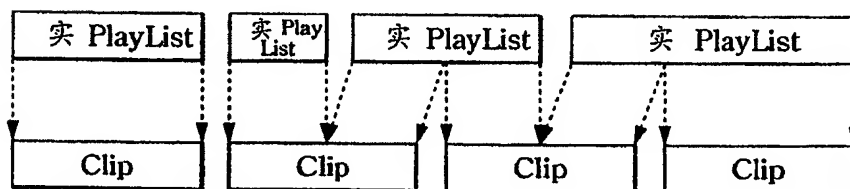


图 24B

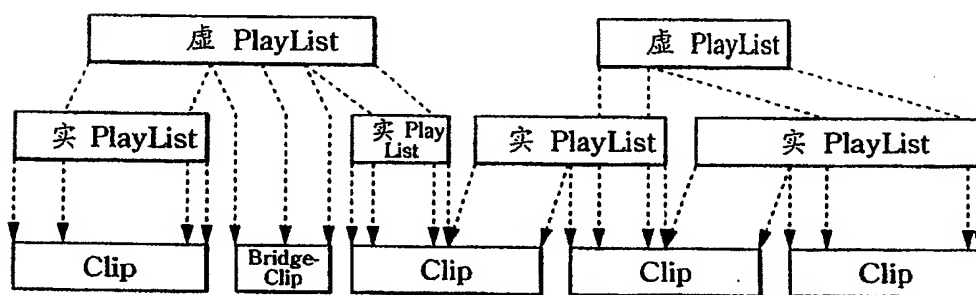


图 24C

句法	字节数	缩写
<b>PlayList()</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>PlayList_type</b>	8	uimsbf
<b>CPI_type</b>	1	bslbf
保留	7	bslbf
<b>UIAppInfoPlayList()</b>		
<b>number_of_PlayItems</b> // 主路径	16	uimsbf
<b>if (&lt;Virtual PlayList&gt;){</b>		
<b>number_of_SubPlayItems</b> // 子路径	16	uimsbf
<b>}else{</b>		
保留	16	bslbf
<b>}</b>		
<b>for (PlayItem_id=0;</b>		
<b>PlayItem_id&lt;number_of_PlayItems;</b>		
<b>PlayItem_id++){</b>		
<b>PlayItem()</b> // 主路径		
<b>}</b>		
<b>if (&lt;Virtual PlayList&gt;){</b>		
<b>if (CPI_type==0 &amp;&amp; PlayList_type==0){</b>		
<b>for (i=0; i&lt;number_of_SubPlayItems; i++){</b>		
<b>SubPlayItem()</b> // 子路径		
<b>}</b>		
<b>}</b>		
<b>}</b>		
<b>}</b>		

图 25

PlayList_type	含义
0	AV记录的播放表; 在该播放表中引用的所有CLIPS一定包含一个或多个视频流
1	音频记录的播放表; 该播放表中引用的所有CLIPS一定包含一个或多个音频流并且一定不包含视频流
2-255	保留

图 26



句法	字节数	缩写
UIAppInfoPlayList20{		
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>PlayList_name</b>	8*256	bslbf
保留	8	bslbf
<b>record_time_and_date</b>	4*14	bslbf
保留	8	bslbf
<b>duration</b>	4*6	bslbf
<b>valid_period</b>	4*8	bslbf
<b>maker_id</b>	16	uimsbf
<b>maker_code</b>	16	uimsbf
保留	11	bslbf
<b>playback_control_flag</b>	1	bslbf
<b>write_protect_flag</b>	1	bslbf
<b>is_played_flag</b>	1	bslbf
<b>archive</b>	2	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
<b>reserved_for_future_use</b>	256	bslbf
}		

图 27

write_protect_flag	含义
0b	PlayList能被自由地删除
1b	除write_protect_flag之外PlayList内容 应不能被擦除和改变

图 28A

is_played_flag	含义
0b	自从其记录开始, PlayList没有被再现
1b	自从其记录开始, PlayList被再现一次

图 28B

文档	含义
00b	没有定义意义
01b	原始的
10b	拷贝
11b	保留

图 28C

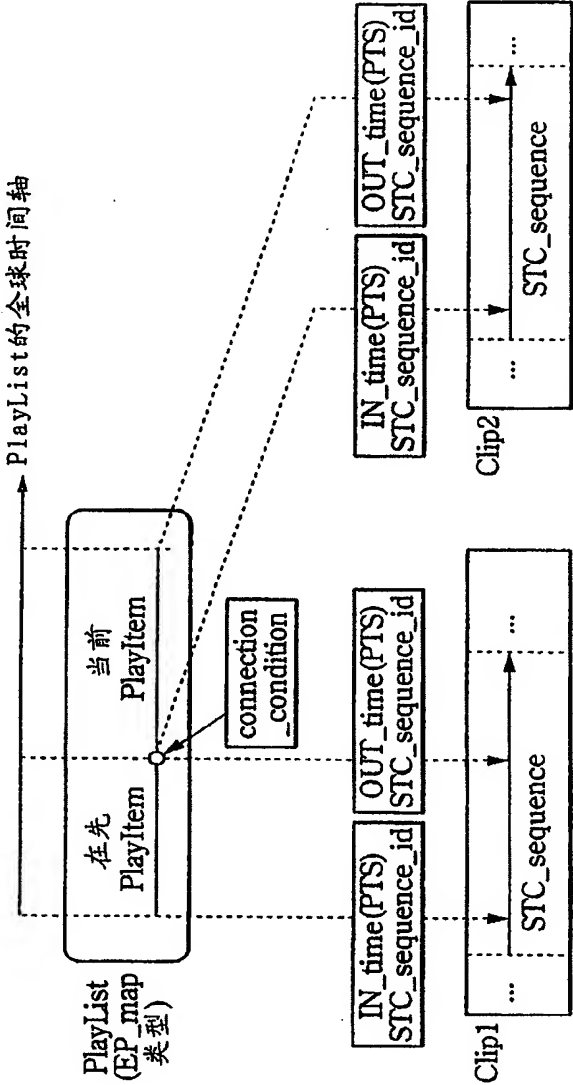


图 29

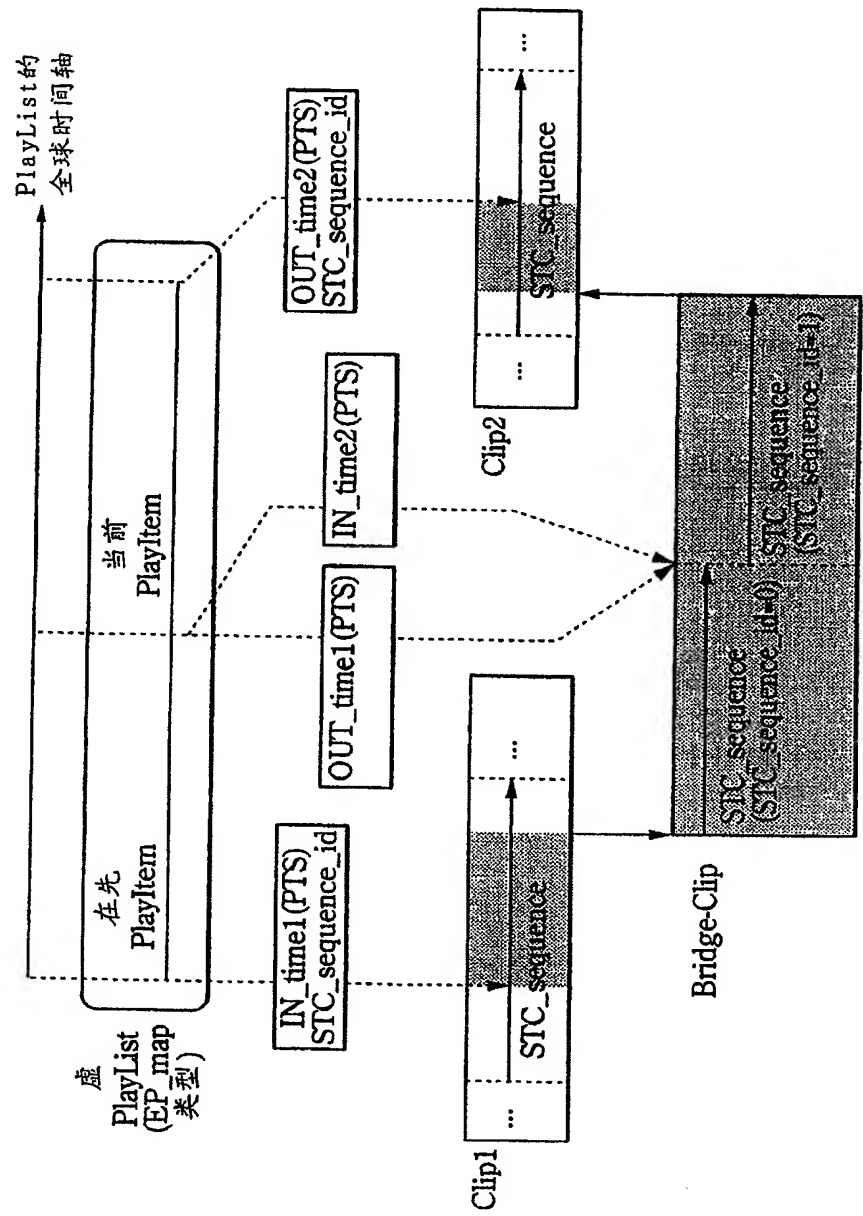


图 30

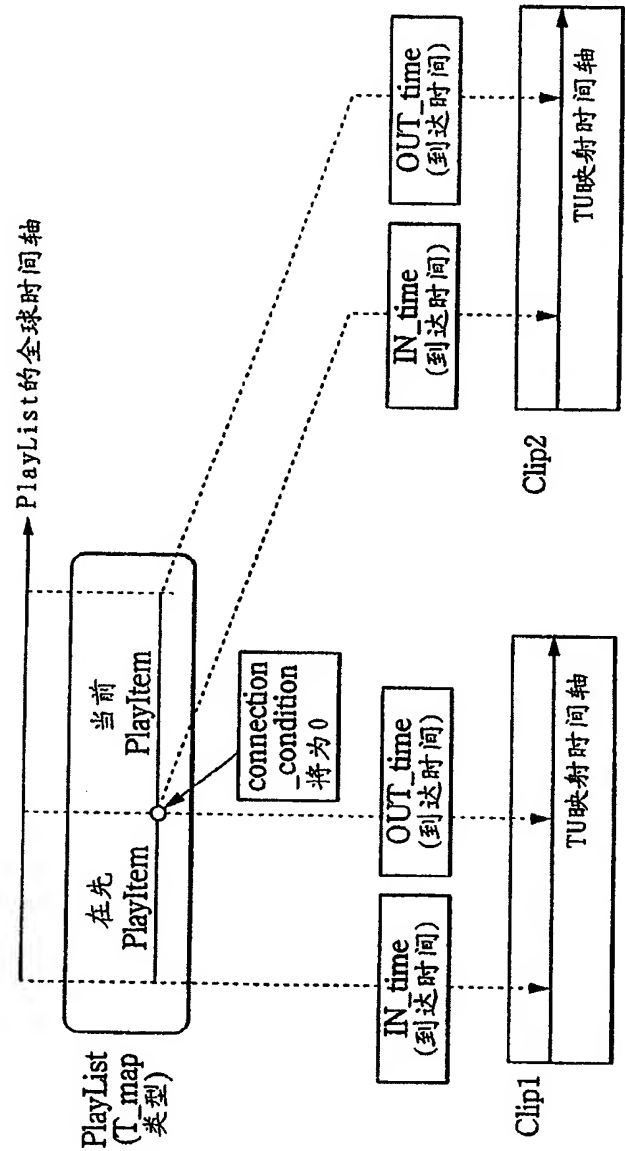


图 31

句法	字节数	缩写
PlayItem(){		
Clip_information_file_name	8*10	bslbf
保留	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
保留	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>){		
if (connection_condition=='10'){		
BridgeSequenceInfo()		
}		
}		
}		

图 32

PlayList () 中的CPI_type	IN_time的语义
EP_map类型	IN_time一定表示对应于PlayItem中的第一显示单元的33位长度的高32位
TU_map类型	IN_time一定是TU_map_time_axis上的时间, 并一定四舍五入到time_unit精度。IN_time是由下述方程计算: $IN\_time = TU\_start\_time \% 2^{32}$

图 33

PlayList () 中的CPI_type	OUT_time的语义
EP_map类型	<p>OUT_time一定表示由下述方程计算的 Presentation_end_TS值的高32位:</p> $\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$ <p>这里PTS_out是对应于PlayItem中的最后显示 单元的33位长的PTS。AU_duration是最后显示 单元的90kHz显示时间。</p>
TU_map类型	<p>OUT_time一定是TU_map_time_axis上的时间， 并四舍五入到time_unit精度。OUT_time由 下式计算:</p> $\text{OUT\_time} = \text{TU\_start\_time} \% 2^{32}$

图 34



connection _condition	含义
00	<ul style="list-style-type: none"> <li>在先PlayItem对当前PlayItem的连接不保证为无缝重新播放。</li> <li>如果Playlist的CPI_type是TU_map类型，该值一定设置在connection_condition中。</li> </ul>
01	<ul style="list-style-type: none"> <li>该状态仅仅当Playlist的CPI_type为EP_map类型时是允许的。</li> <li>因为系统时基的非连续点(STC基)，当前PlayItem和在先PlayItem表示分开。</li> </ul>
10	<ul style="list-style-type: none"> <li>该状态仅仅当Playlist的CPI_type是EP_map类型时是允许的。</li> <li>该状态仅对虚Playlist是允许的。</li> <li>在先PlayItem对当前PlayItem的连接被保证为无缝重新播放。</li> <li>在先PlayItem使用BridgeSequence连接到当前PlayItem。DVR MPEG-2传输流一定遵守后面说明的DVR-STD。</li> </ul>
11	<ul style="list-style-type: none"> <li>该状态仅当Playlist的CPI_type是EP_map类型时是允许的。</li> <li>在先PlayItem到当前PlayItem的连接保证为无缝重新播放。</li> <li>在先PlayItem在不使用BridgeSequence的情况下连接到当前PlayItem。DVR MPEG-2传输流一定遵守后面说明的DVR-STD。</li> </ul>

图 35

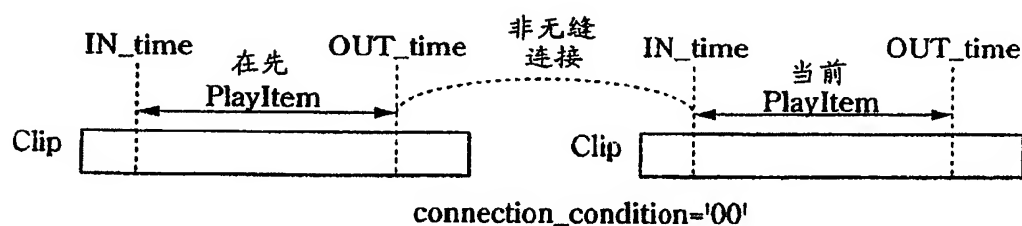


图 36A

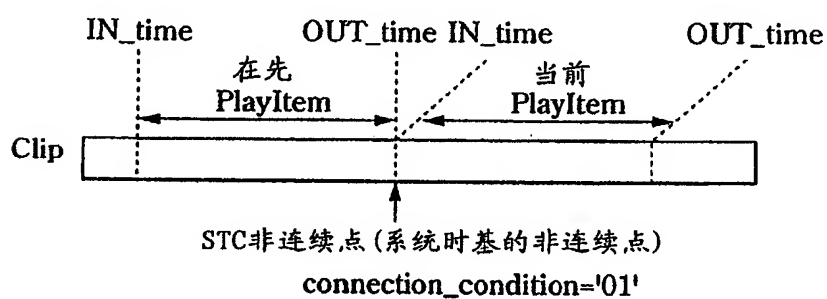


图 36B

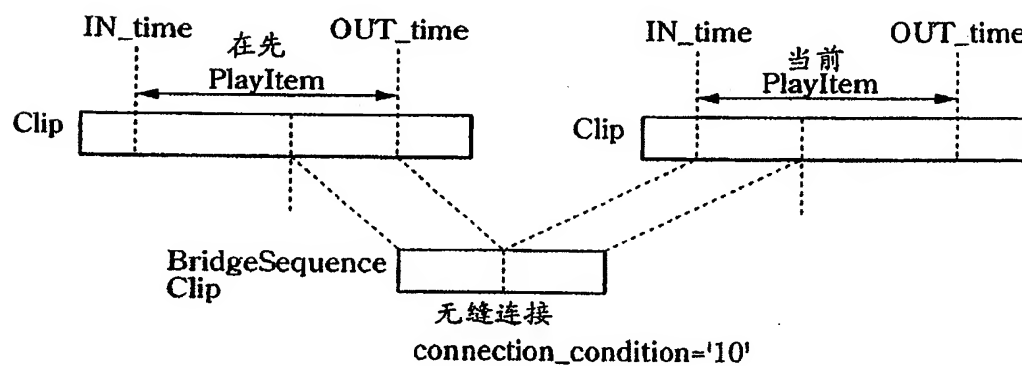


图 36C

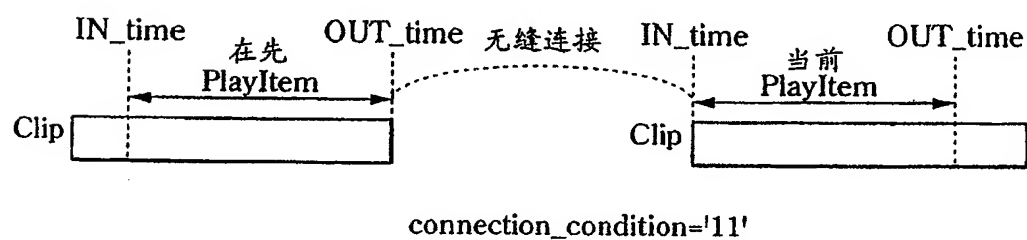


图 36D

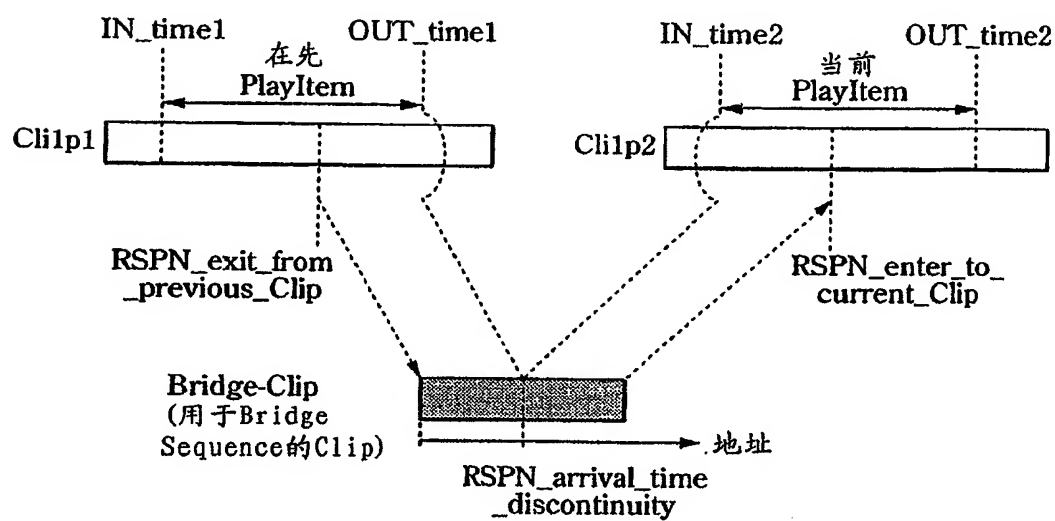


图 37

句法	字节数	缩写
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

图 38

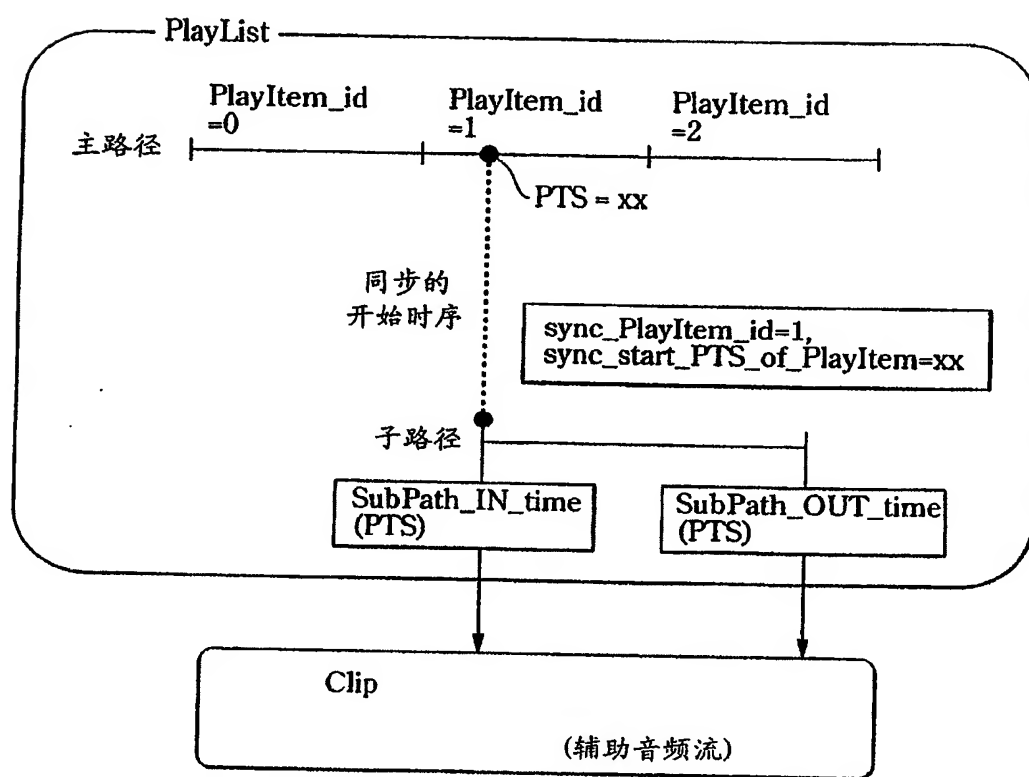


图 39

句法	字节数	缩写
SubPlayItem(){		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

图 40

SubPath_type	含义
0x00	辅助音频流路径
0x01-0xff	保留

图 41

句法	字节数	缩写
PlayListMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayList_marks</b>	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
保留	8	bslbf
<b>mark_type</b>	8	bslbf
<b>mark_time_stamp</b>	32	uimsbf
<b>PlayItem_id</b>	8	uimsbf
保留	24	uimsbf
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>mark_name</b>	8*256	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

图 42



Mark_type	含义	说明
0x00	恢复-标记	重新播放恢复点。在PlayListMark()中定义的重新播放恢复点数一定为0或1。
0x01	书-标记	重新播放PlayList的入口点。该标记能够由用户设置和用作为指定喜爱的场景开始点的标记。
0x02	跳步-标记	跳步标记点。播放器从该点到程序结束跳过程序。在PlayListMark()中定义的跳步标记点数一定是0或1。
0x03-0x8F	保留	
0x90-0xFF	保留	保留用于ClipMark()

图 43

PlayList () 中的CPI_type	mark_time_stamp的语义
EP_map类型	mark_time_stamp一定表示对应于由标记 引用的表示单元的33位长度PTS的32位
TU_map类型	mark_time_stamp一定是关于TU_map_time_axis 的时间并且一定四舍五入到time_unit精度。 mark_time_stamp通过下述方程计算:  $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

图 44

句法	字节数	缩写
<b>zzzzz.clpi {</b>		
<b>STC_Info_Start_address</b>	32	uimbsf
<b>ProgramInfo_Start_address</b>	32	uimbsf
<b>CPI_Start_address</b>	32	uimbsf
<b>ClipMark_Start_address</b>	32	uimbsf
<b>MakersPrivateData_Start_address</b>	32	uimbsf
保留	96	bslbf
<b>ClipInfo()</b>		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
<b>STC_Info()</b>		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
<b>ProgramInfo()</b>		
for (i=0;i<N3;i++){		
padding_word	16	bslbf
}		
<b>CPI()</b>		
for (i=0;i<N4;i++){		
padding_word	16	bslbf
}		
<b>ClipMark()</b>		
for (i=0;i<N5;i++){		
padding_word	16	bslbf
}		
<b>MakersPrivateData()</b>		
}		

图 45

句法	字节数	缩写
ClipInfo(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>Clip_stream_type</b>	8	bslbf
<b>offset_SPN</b>	32	uimsbf
<b>TS_recording_rate</b>	24	uimsbf
保留	8	bslbf
<b>record_time_and_date</b>	4*14	bslbf
保留	8	bslbf
<b>duration</b>	4*6	bslbf
保留	7	bslbf
<b>time_controlled_flag</b>	1	bslbf
<b>TS_average_rate</b>	24	uimsbf
<i>if (Clip_stream_type==1) // Bridge-Clip AV stream</i>		
<b>RSPN_arrival_time_discontinuity</b>	32	uimsbf
else		
保留	32	bslbf
<b>reserved_for_system_use</b>	144	bslbf
保留	11	bslbf
<b>is_format_identifier_valid</b>	1	bslbf
<b>is_original_network_ID_valid</b>	1	bslbf
<b>is_transport_stream_ID_valid</b>	1	bslbf
<b>is_service_ID_valid</b>	1	bslbf
<b>is_country_code_valid</b>	1	bslbf
<b>format_identifier</b>	32	bslbf
<b>original_network_ID</b>	16	uimsbf
<b>transport_stream_ID</b>	16	uimsbf
<b>service_ID</b>	16	uimsbf
<b>country_code</b>	24	bslbf
<b>stream_format_name</b>	16*8	bslbf
<b>reserved_for_fortune_use</b>	256	bslbf
}		

图 46

Clip_stream_type	含义
0	Clip AV流
1	Bridge-Clip AV流
2-255	保留

图 47

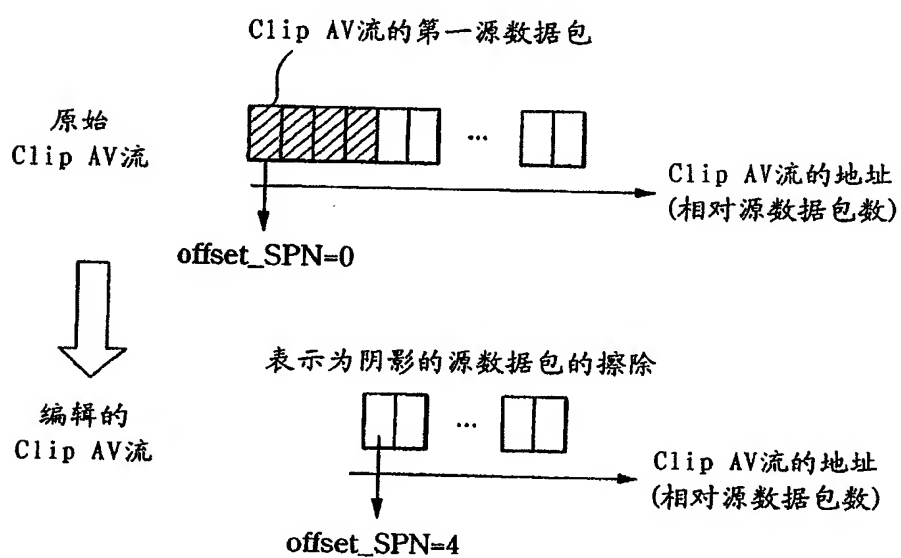


图 48

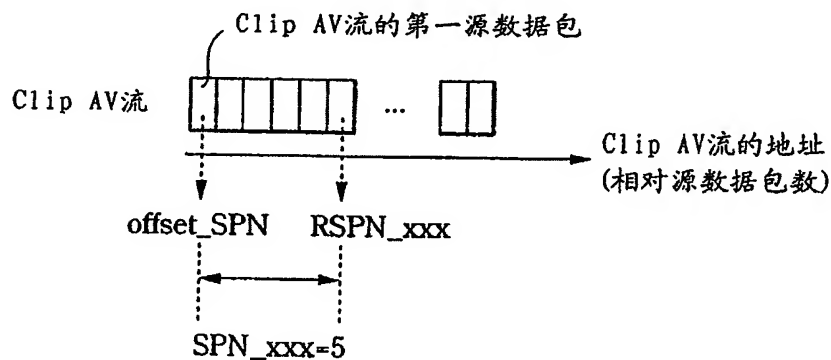


图 49

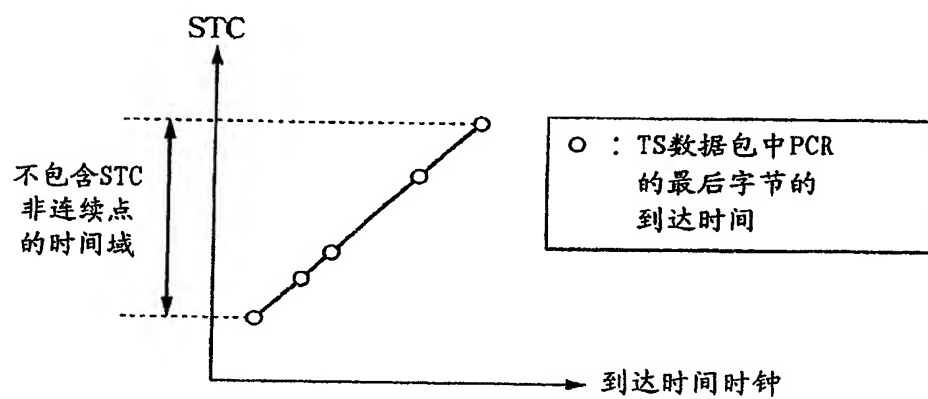


图 50A

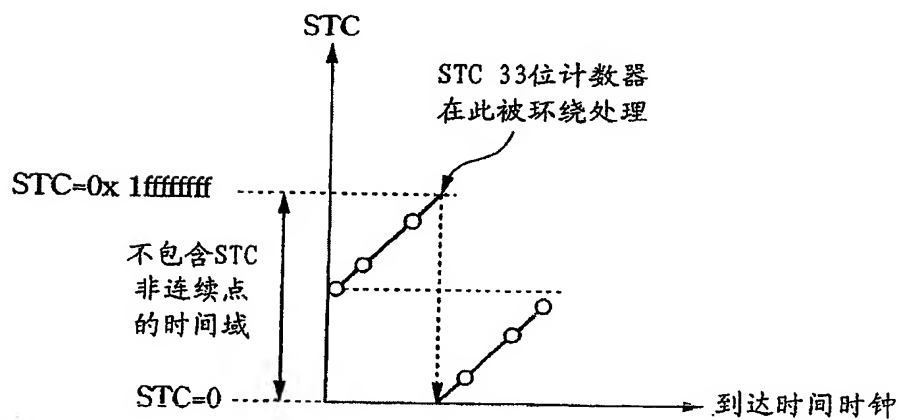


图 50B

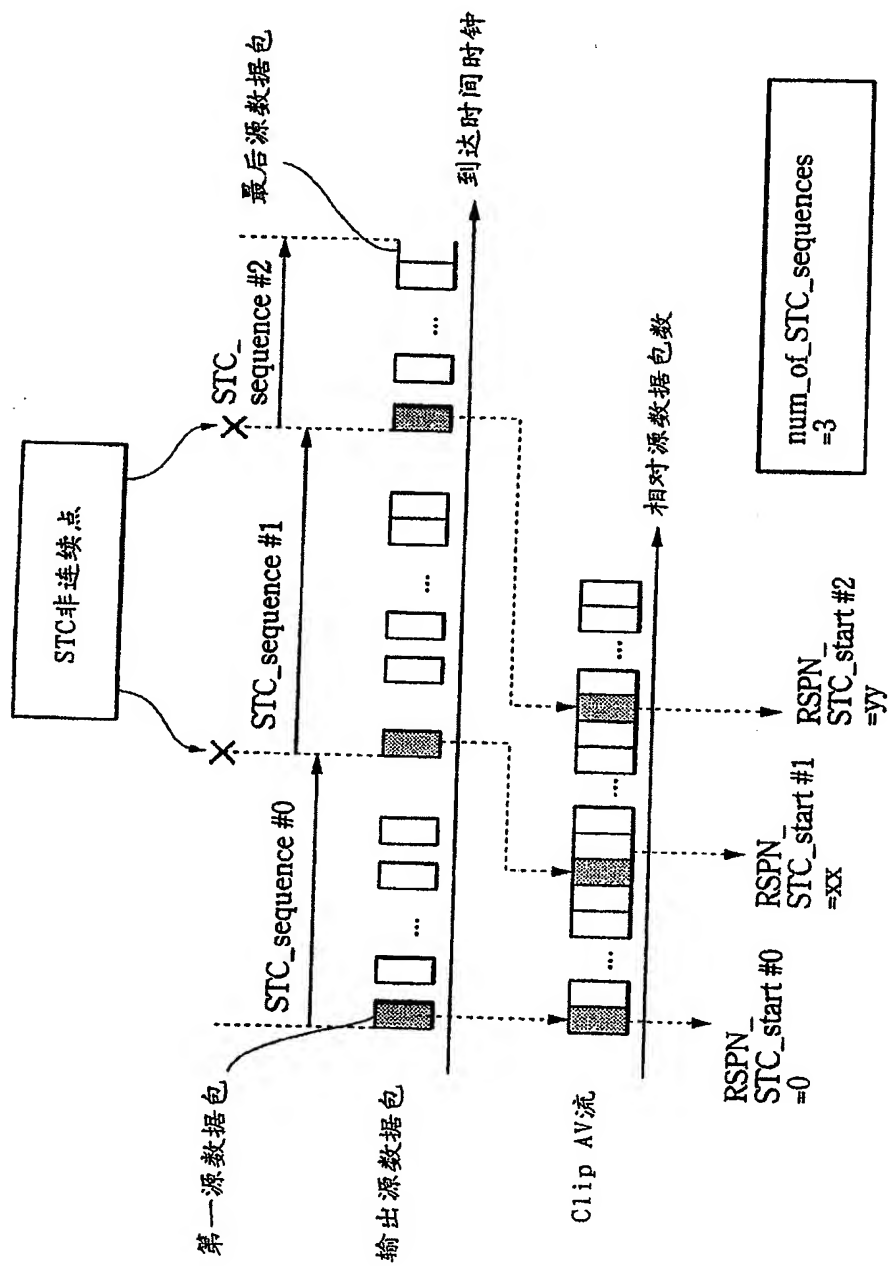


图 51



句法	字节数	缩写
<b>STC_Info()</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>if (length !=0){</b>		
保留	8	bslbf
<b>num_of_STC_sequences</b>	8	uimsbf
<b>for (STC_sequence_id=0;</b> <b>STC_sequence_id&lt;num_of_STC_sequences;</b> <b>STC_sequence_id++){</b>		
保留	32	bslbf
<b>RSPN_STC_start</b>	32	uimsbf
<b>}</b>		
<b>}</b>		
<b>}</b>		

图 52

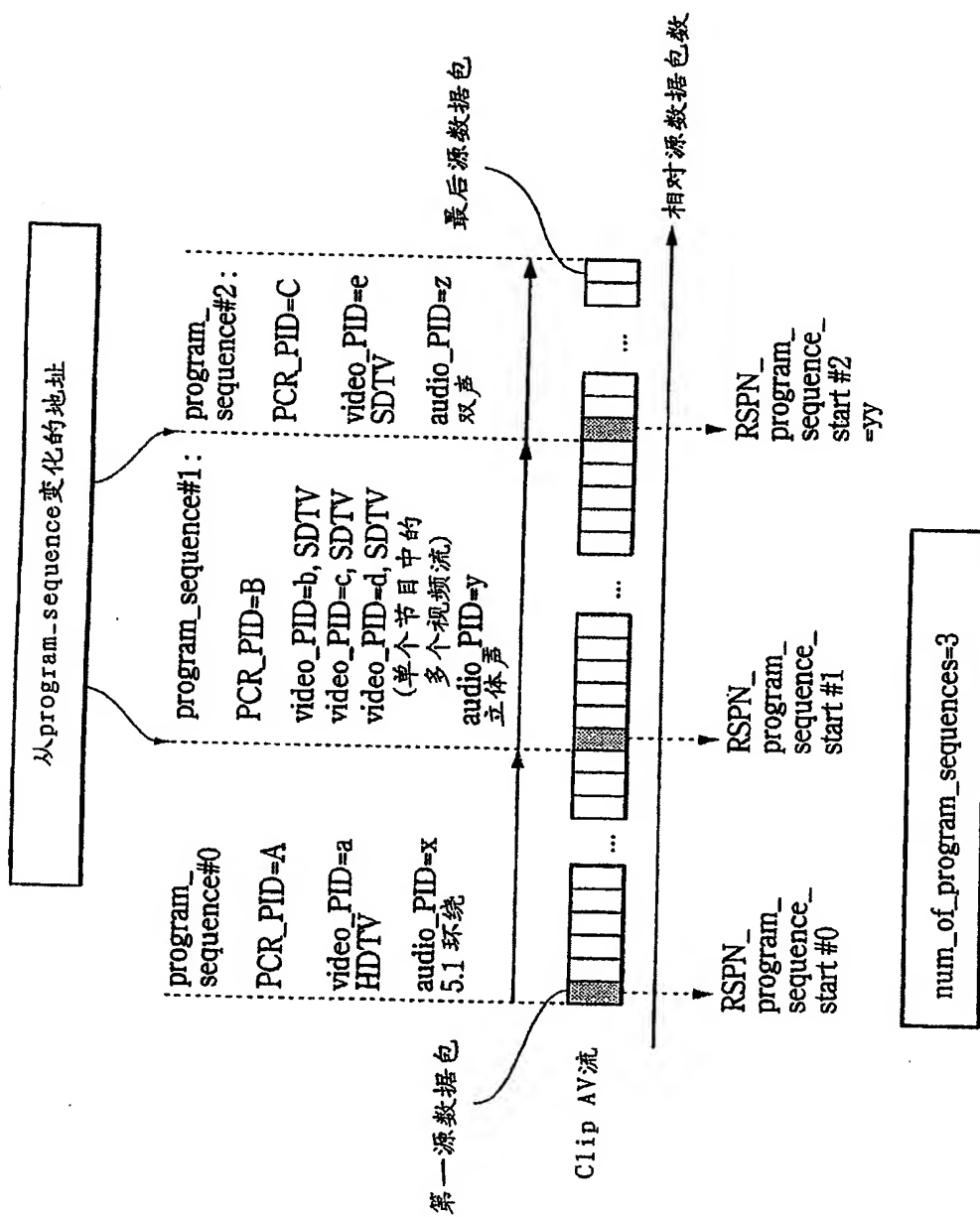


图 53

句法	字节数	缩写
<b>ProgramInfo()</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>if (length !=0){</b>		
保留	8	bslbf
<b>number_of_program_sequences</b>	8	uimsbf
<b>for (i=0;i&lt;number_of_program_sequences;i++){</b>		
<b>RSPN_program_sequence_start</b>	32	uimsbf
保留	48	bslbf
<b>PCR_PID</b>	16	bslbf
<b>number_of_videos</b>	8	uimsbf
<b>number_of_audios</b>	8	uimsbf
<b>for (k=0;k&lt;number_of_videos;k++){</b>		
<b>video_stream_PID</b>	16	bslbf
<b>VideoCodingInfo()</b>		
<b>}</b>		
<b>for (k=0;k&lt;number_of_audios;k++){</b>		
<b>audio_stream_PID</b>	16	bslbf
<b>AudioCodingInfo()</b>		
<b>}</b>		
<b>}</b>		
<b>}</b>		

图 54

句法	字节数	缩写
VideoCodingInfo() {		
<b>video_format</b>	8	uimsbf
<b>frame_rate</b>	8	uimsbf
<b>display_aspect_ratio</b>	8	uimsbf
保留	8	bslbf
}		

图 55

video_format	含义
0	480i
1	576i
2	480p (包括640×480p格式)
3	1080i
4	720p
5	1080p
6-254	保留
255	无信息

图 56

frame_rate	含义
0	禁止
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	保留
255	无信息

图 57

display_aspect_ratio	含义
0	禁止
1	保留
2	4: 3显示纵横比
3	16: 9显示纵横比
4-254	保留
255	无信息

图 58

句法	字节数	缩写
AudioCodingInfo() {		
<b>audio_format</b>	8	uimsbf
<b>audio_component_type</b>	8	uimsbf
<b>sampling_frequency</b>	8	uimsbf
保留	8	bslbf
}		

图 59

audio_coding	含义
0	MPEG-1 音频层I或II
1	杜比AC-3音频
2	MPEG-2 AAC
3	MPEG-2多声道音频, 与MPEG-1兼容的反向
4	SESF LPCM音频
5-254	保留
255	无信息

图 60



audio_component_type	含义
0	单声道
1	双声道
2	立体声 (2声道)
3	多舌音, 多声道
4	环绕声
5	视觉损坏的音频说明
6	难听的音频
7-254	保留
255	无信息

图 61

sampling_frequency	含义
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	保留
255	无信息

图 62

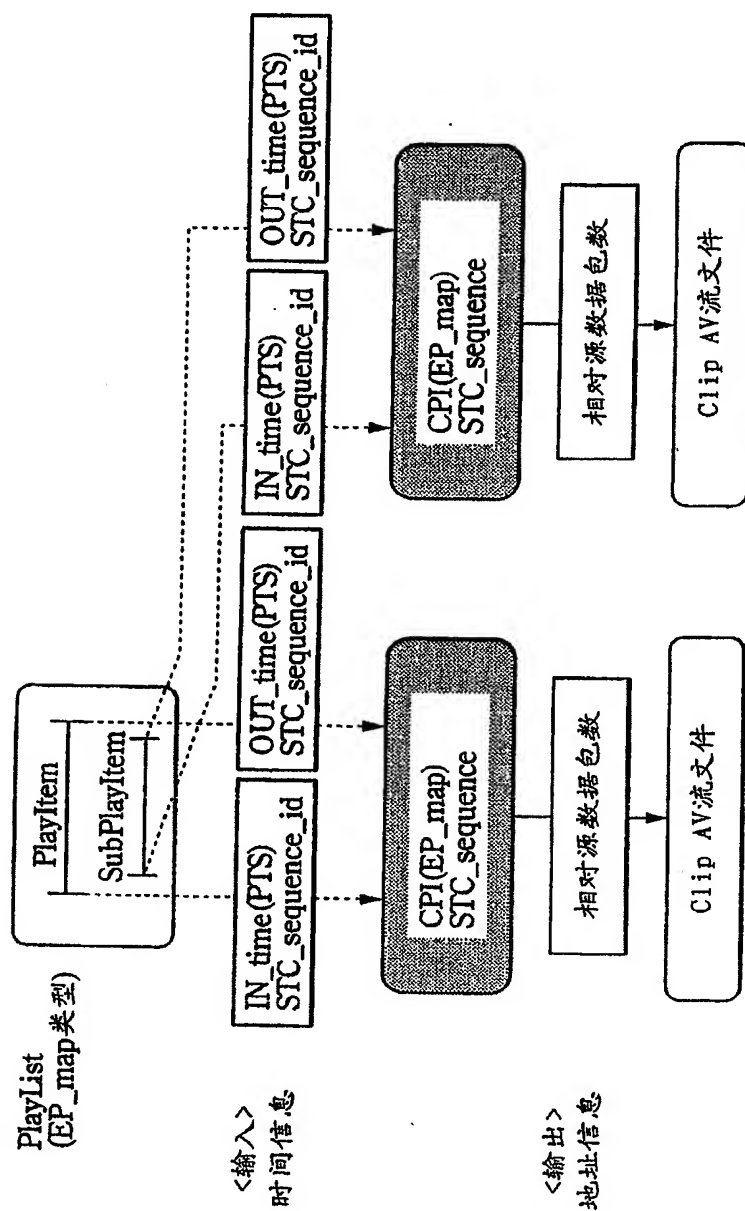


图 63

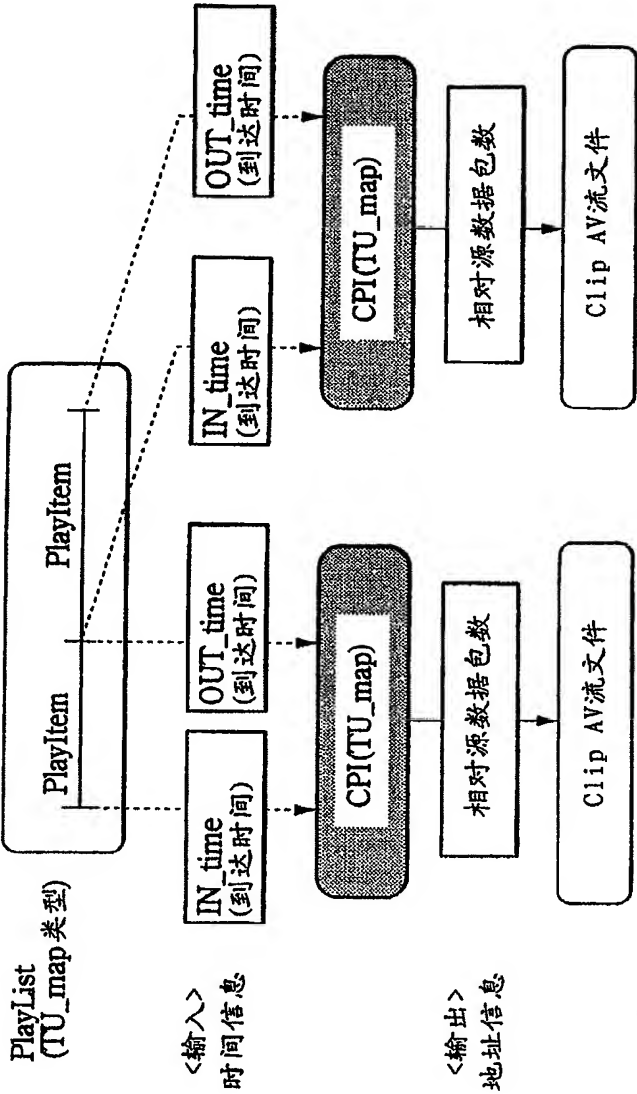


图 64

句法	字节数	缩写
CPI(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
保留	15	bslbf
<b>CPI_type</b>	1	bslbf
if (CPI_type==0)		
<b>EP_map()</b>		
else		
<b>TU_map()</b>		
}		

图 65

CPI_type	含义
0	EP映射类型
1	TU映射类型

图 66

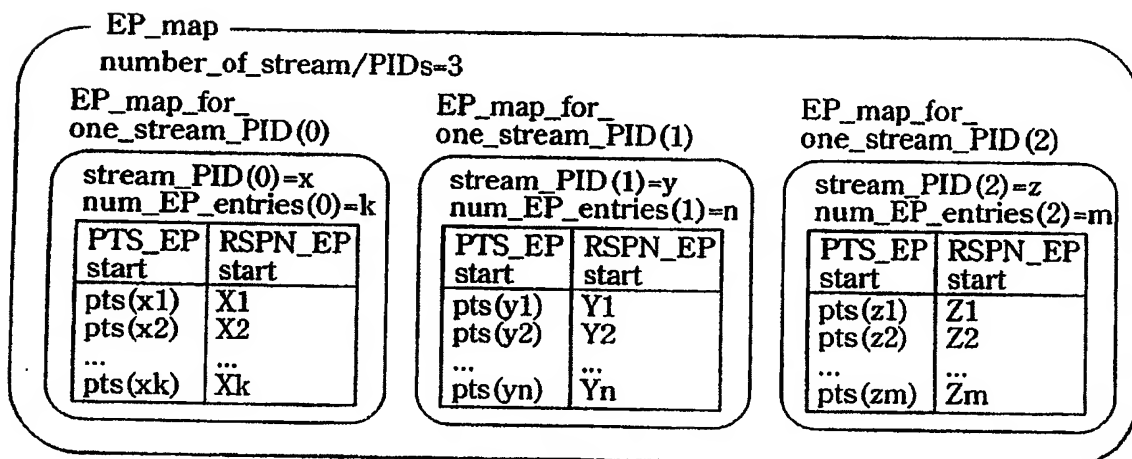
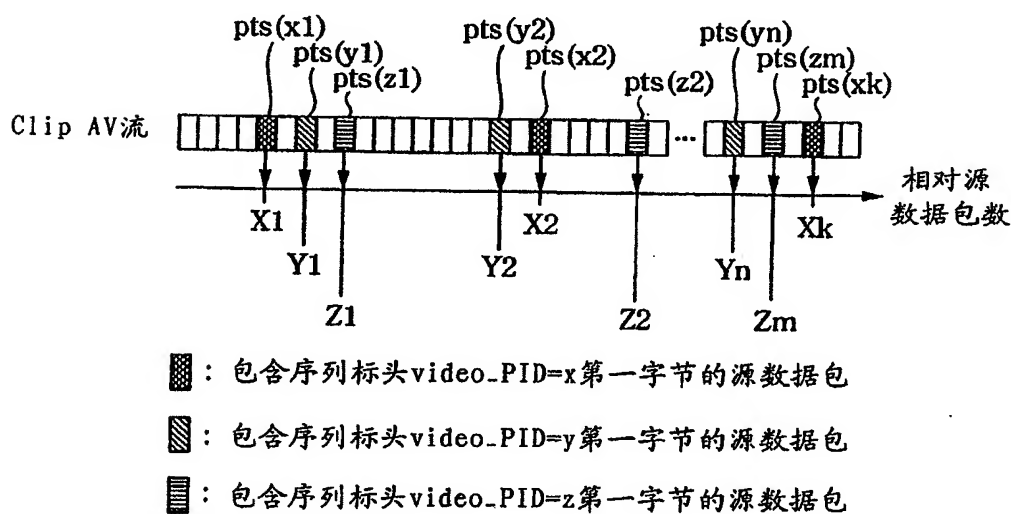
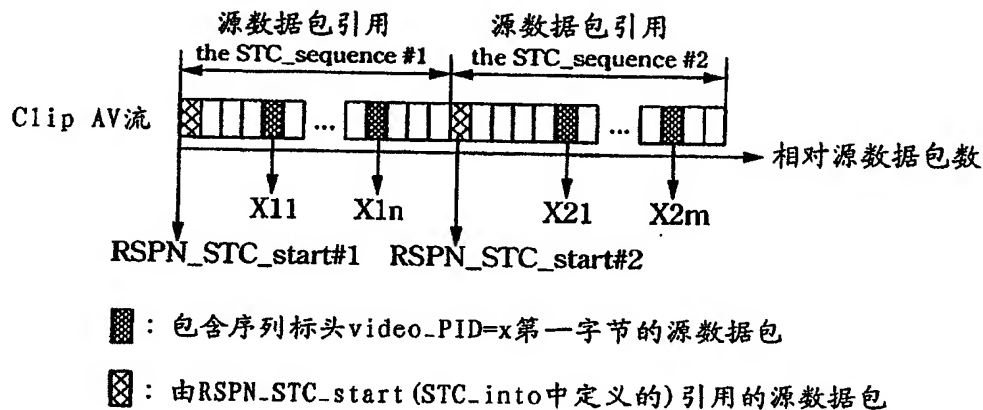


图 67



EP\_map\_for\_one\_stream\_PID  
video\_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11	属于STC_sequence#1的数据
...	...	
pts(x1n)	X1n	
pts(x21)	X21	属于STC_sequence#2的数据
...	...	
pts(x2m)	X2m	

边界

RSPN\_STC\_start #2 < X21

图 68

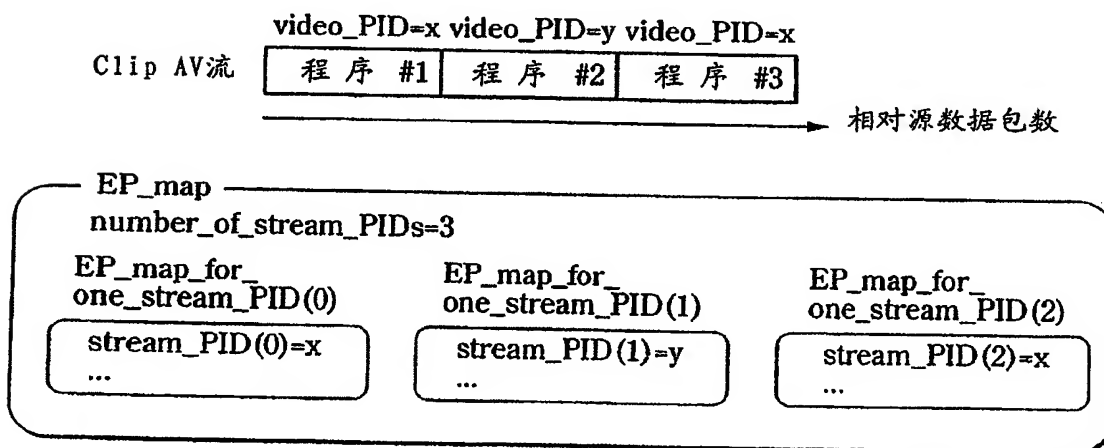


图 69

句法	字节数	缩写
EP_map0{		
保留	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

图 70

EP_type	含义
0	视频
1	音频
2-15	保留

图 71



句法	字节数	缩写
EP_map_for_one_stream_PID(N) {		
for (i=0;i<N;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

图 72

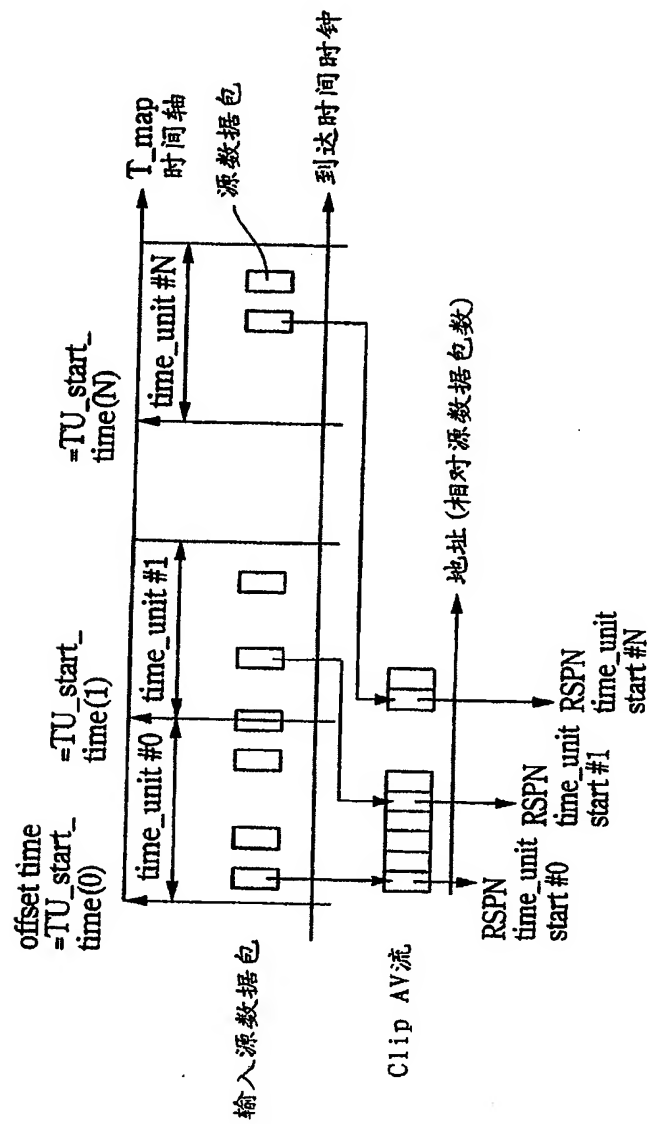


图 73

句法	字节数	缩写
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

图 74

句法	字节数	缩写
<b>ClipMark0{</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
<b>for (i=0; i&lt;number_of_clip_marks; i++){</b>		
保留	8	bslbf
<b>mark_type</b>	8	bslbf
<b>mark_time_stamp</b>	32	uimsbf
<b>STC_sequence_id</b>	8	uimsbf
保留	24	bslbf
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>mark_name</b>	8*256	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
<b>}</b>		
<b>}</b>		

图 75

Mark_type	含义	注释
0x00-0x8F	保留	保留用于PlayListMark()
0x90	事件开始标记	说明节目开始点的标记点
0x91	本地事件开始标记	说明节目中本地场景的标记点
0x92	场景开始标记	表示场景变化点的标记点
0x93-0xFF	保留	

图 76

PlayList 0 中的CPI_type	mark_time_stamp的语义
EP_map类型	mark_time_stamp一定表示对应于由标记 引用的表示单元的33位长度PTS的高32位
TU_map类型	mark_time_stamp一定是TU_map_time_axis 上的时间并一定四舍五入到time_unit精度。 mark_time_stamp由下述方程计算：  $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

图 77

句法	字节数	缩写
ClipMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
<b>reserved</b>	8	bslbf
<b>mark_type</b>	8	bslbf
<b>reserved_for_MakerID</b>	16	bslbf
<b>mark_entry()</b>		
<b>representative_picture_entry()</b>		
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

图 78

Mark_type	含义	备注
0x00-0x8F	保留	为PlayListMark()保留
0x90	事件开始标记	指示程序开始点的标记点
0x91	本地事件开始标记	指示程序中本地场景的标记点
0x92	场景开始标记	指示场景开始点的标记点
0x93	场景结束标记	指示场景结束点的标记点
0x94	CM开始标记	指示CM开始点的标记点
0x95	CM结束标记	指示CM结束点的标记点
0x96-0xBF	为ClipMark的未来扩展保留DVR格式	
0xC0-0xFF	可分配给用于制造商唯一应用的标记	

图 79

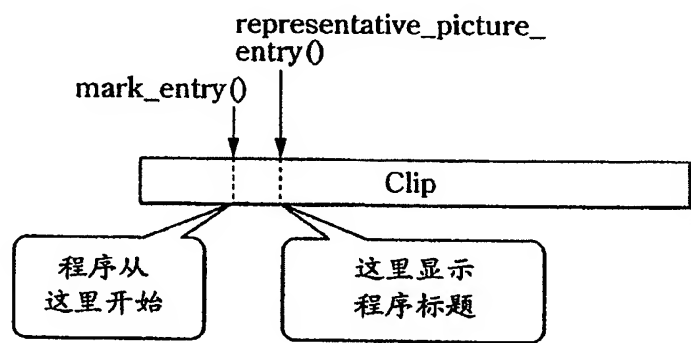


图 80

句法	字节数	缩写
<b>mark_entry()/representative_picture_entry(){</b>		
<b>mark_time_stamp</b>	32	uimsbf
<b>STC_sequence_id</b>	8	uimsbf
<b>reserved</b>	24	bslbf
<b>}</b>		

图 81

句法	字节数	缩写
<b>mark_entry()/representative_picture_entry(){</b>		
<b>RSPN_ref_EP_start</b>	32	uimsbf
<b>offset_num_pictures</b>	32	uimsbf
<b>}</b>		

图 82



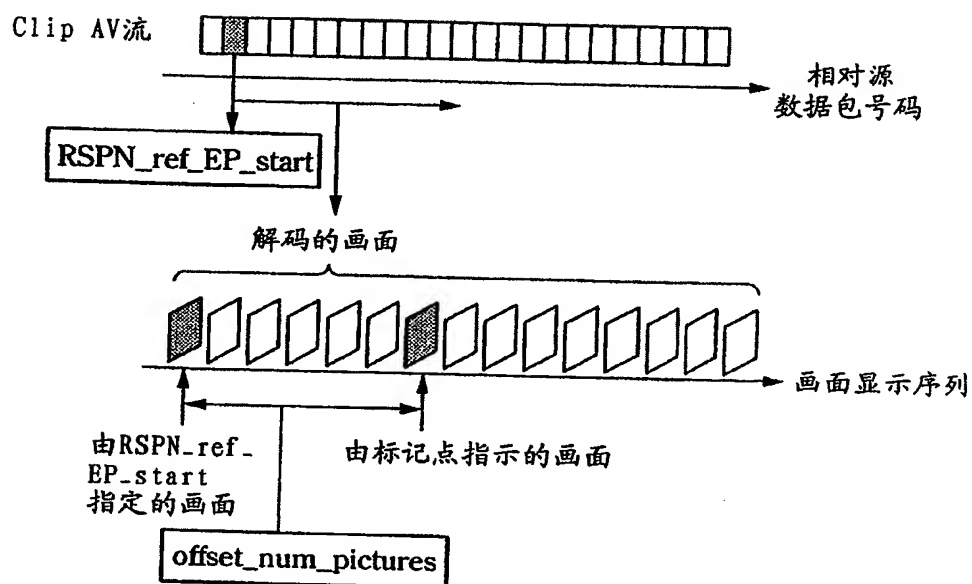


图 83

句法	字节数	缩写
mark_entry()/representative_picture_entry(){		
RSPN_mark_point	32	uimsbf
}		

图 84

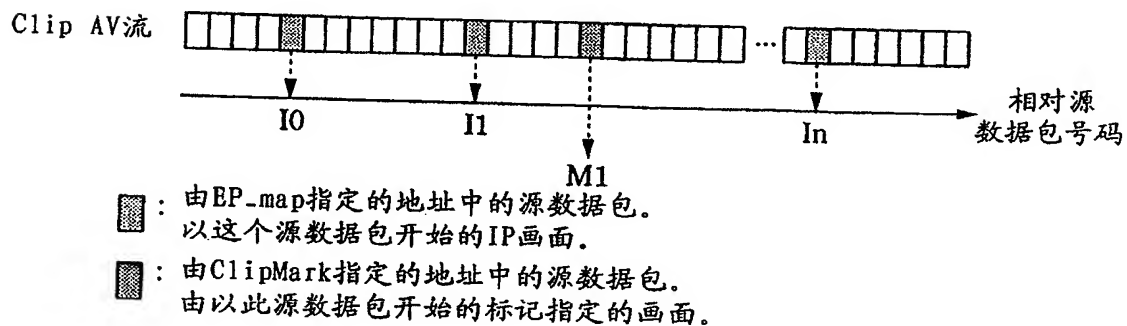


图 85

句法	字节数	缩写
menu.thmb/mark.thmb() {		
reserved	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

图 86

句法	字节数	缩写
Thumbnail(){		
<b>version_number</b>	8*4	char
<b>length</b>	32	uimsbf
if (length !=0){		
<b>tn_blocks_start_address</b>	32	bslbf
<b>number_of_thumbnails</b>	16	uimsbf
<b>tn_block_size</b>	16	uimsbf
<b>number_of_tn_blocks</b>	16	uimsbf
<b>reserved</b>	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
<b>thumbnail_index</b>	16	uimsbf
<b>thumbnail_picture_format</b>	8	bslbf
<b>reserved</b>	8	bslbf
<b>picture_data_size</b>	32	uimsbf
<b>start_tn_block_number</b>	16	uimsbf
<b>x_picture_length</b>	16	uimsbf
<b>y_picture_length</b>	16	uimsbf
<b>reserved</b>	16	uimsbf
}		
<b>stuffing_bytes</b>	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
<b>tn_block</b>	tn_block_size*1024*8	
}		
}		
}		

图 87

Thumbnail_picture_format	含义
0x00	MPEG-2视频I画面
0x01	DCF(约束的JPEG)
0x02	PNG
0x03-0xff	保留

图 88

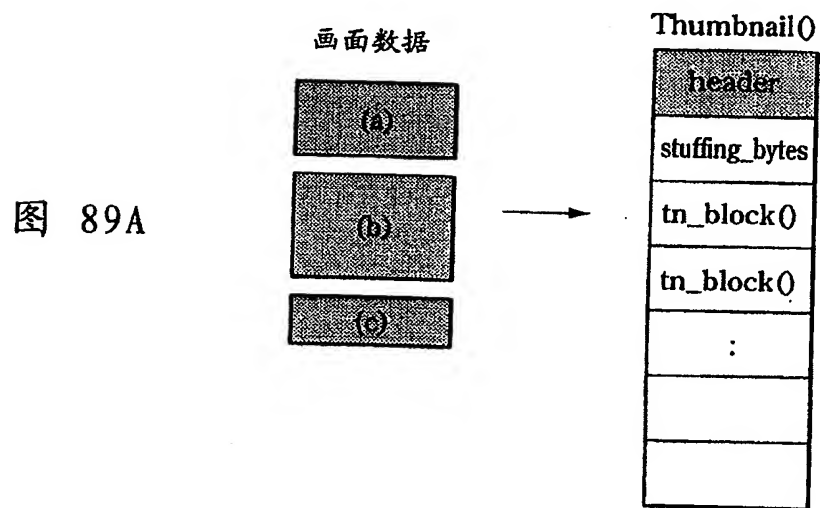
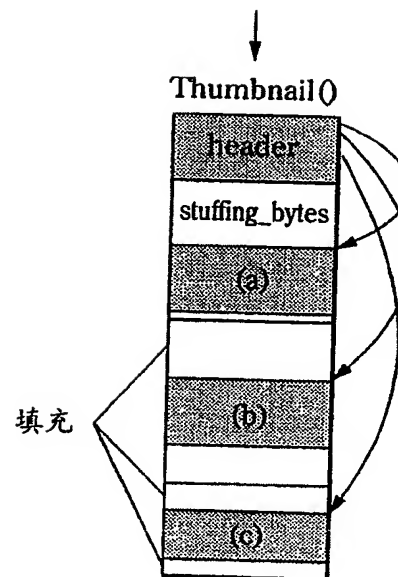


图 89B



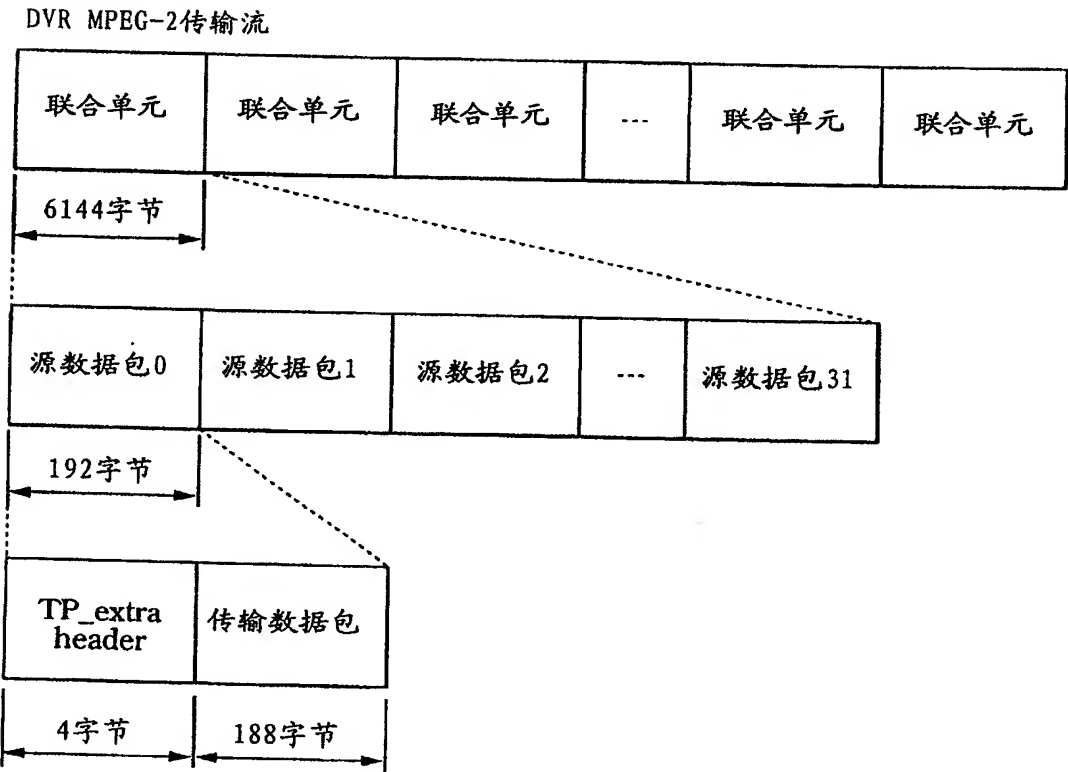


图 90

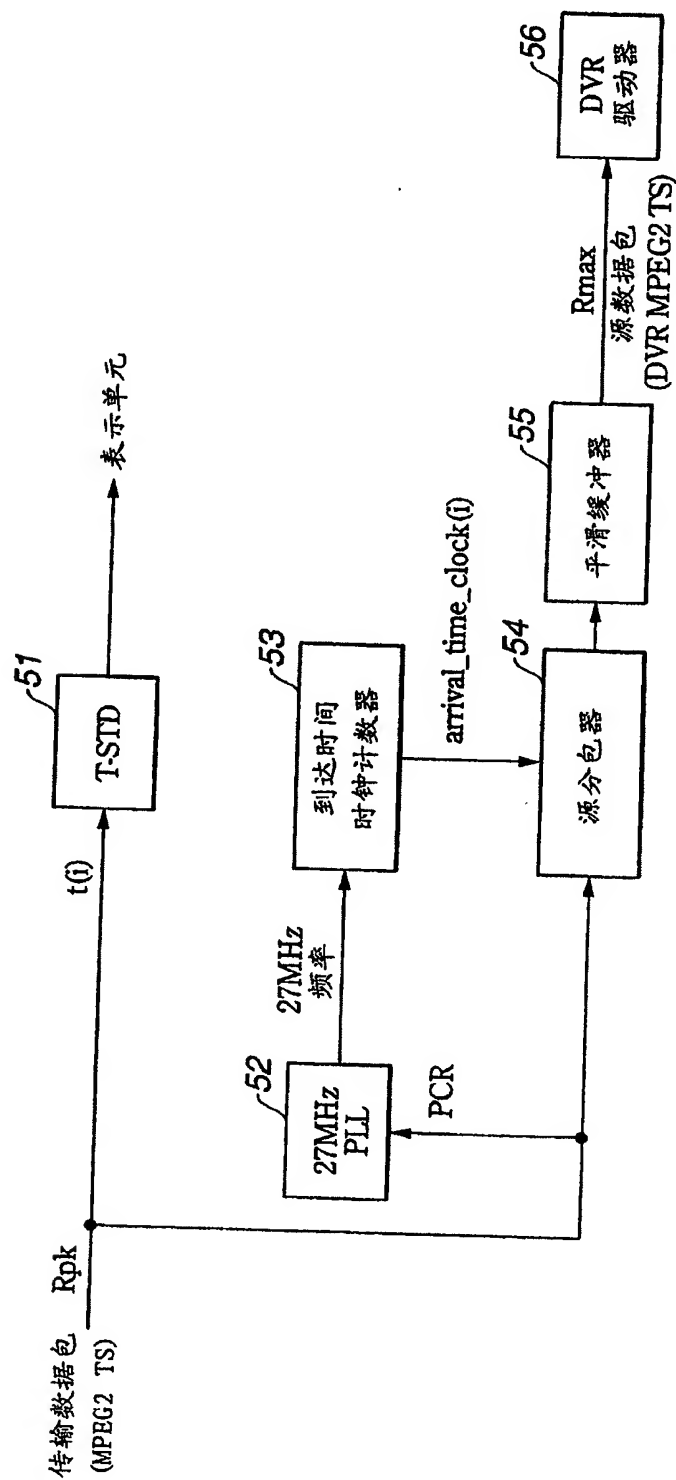


图 91

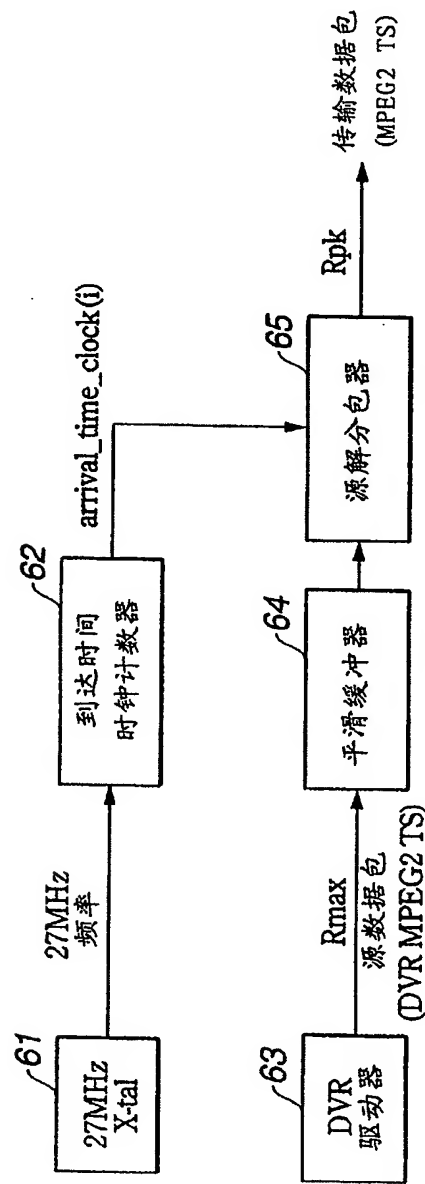


图 92

句法	字节数	缩写
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

图 93



句法	字节数	缩写
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

图 94

copy_permission _indicator	含义
00	自由复制
01	不再复制
10	复制一次
11	禁止复制

图 95

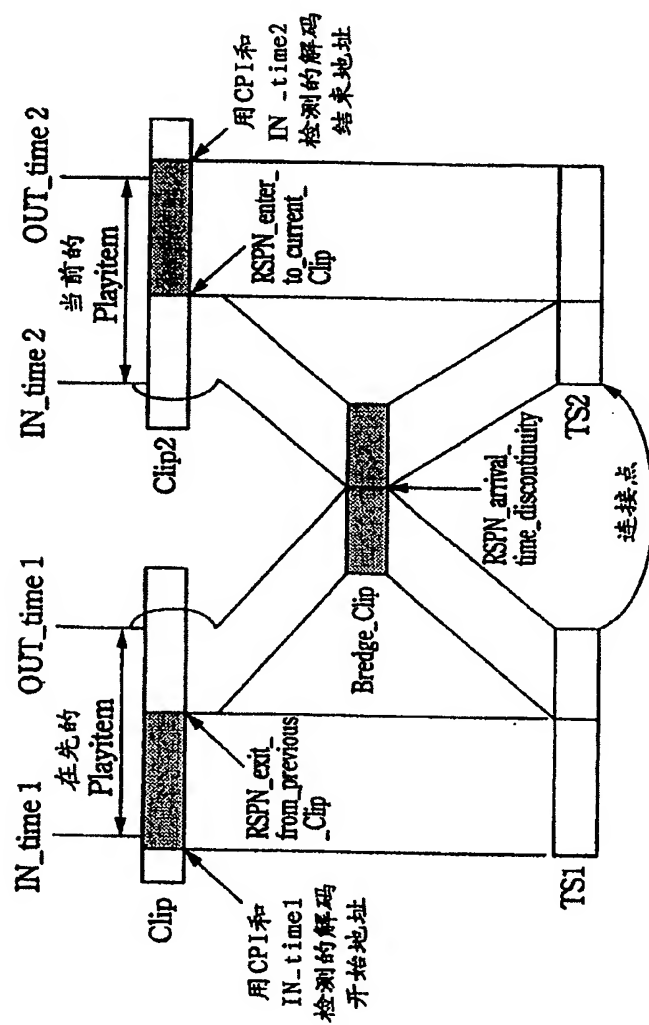


图 96

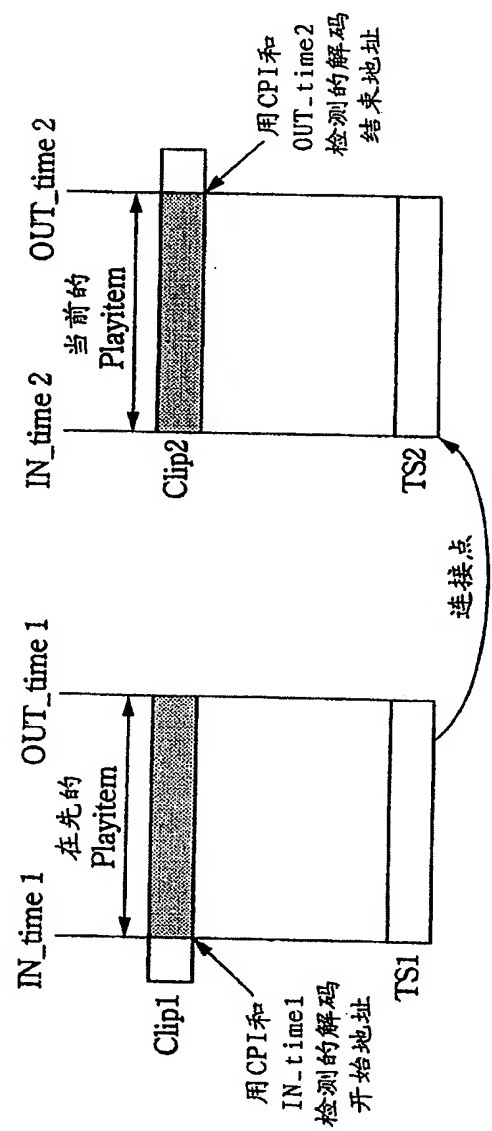


图 97

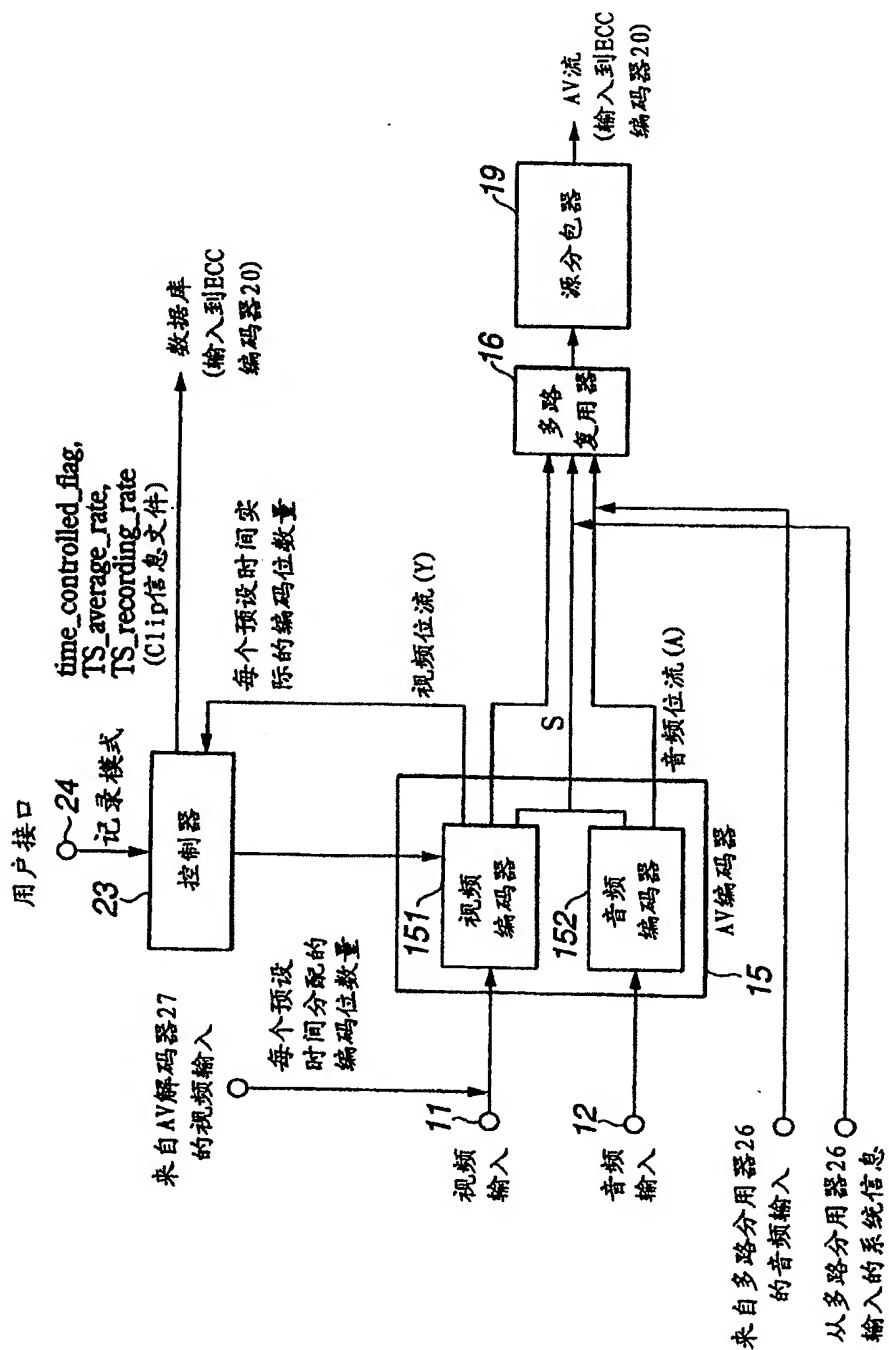


图 98

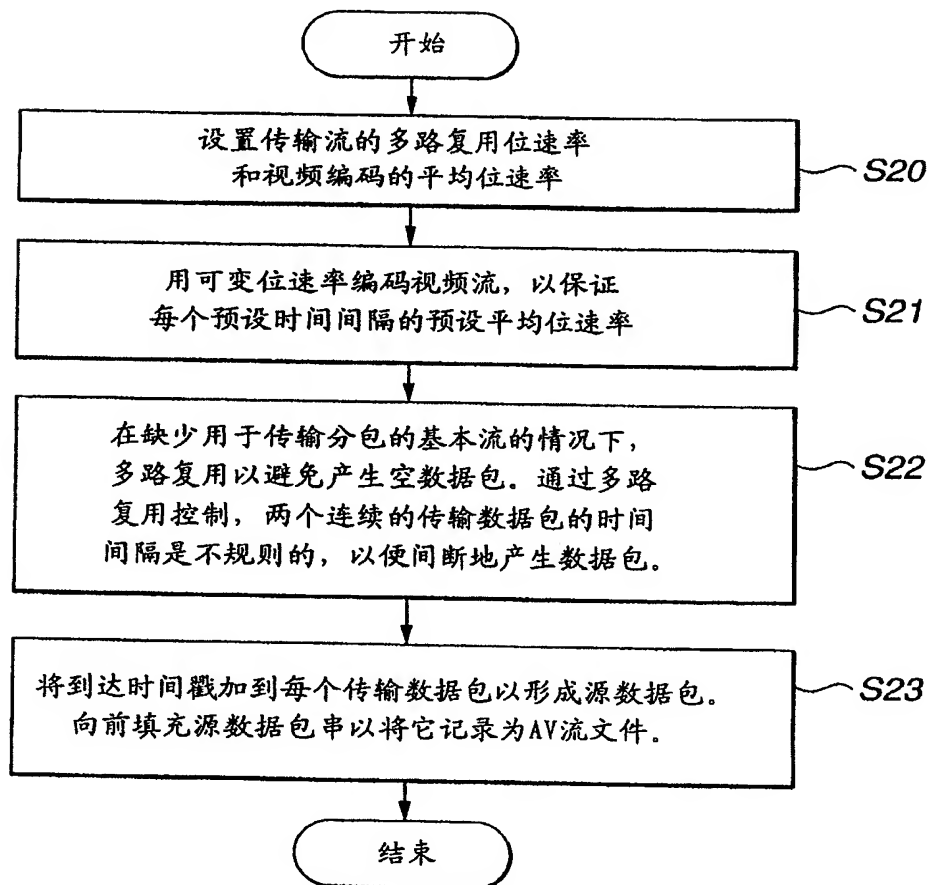


图 99

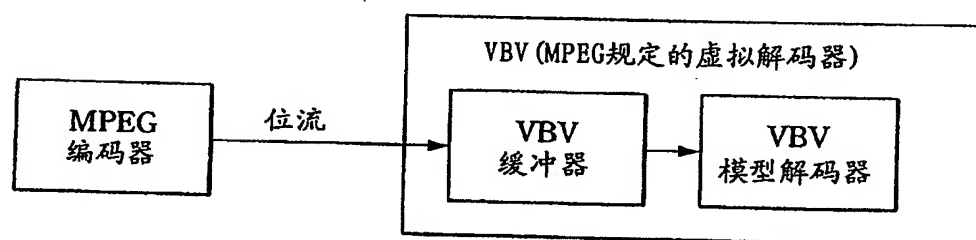


图 100

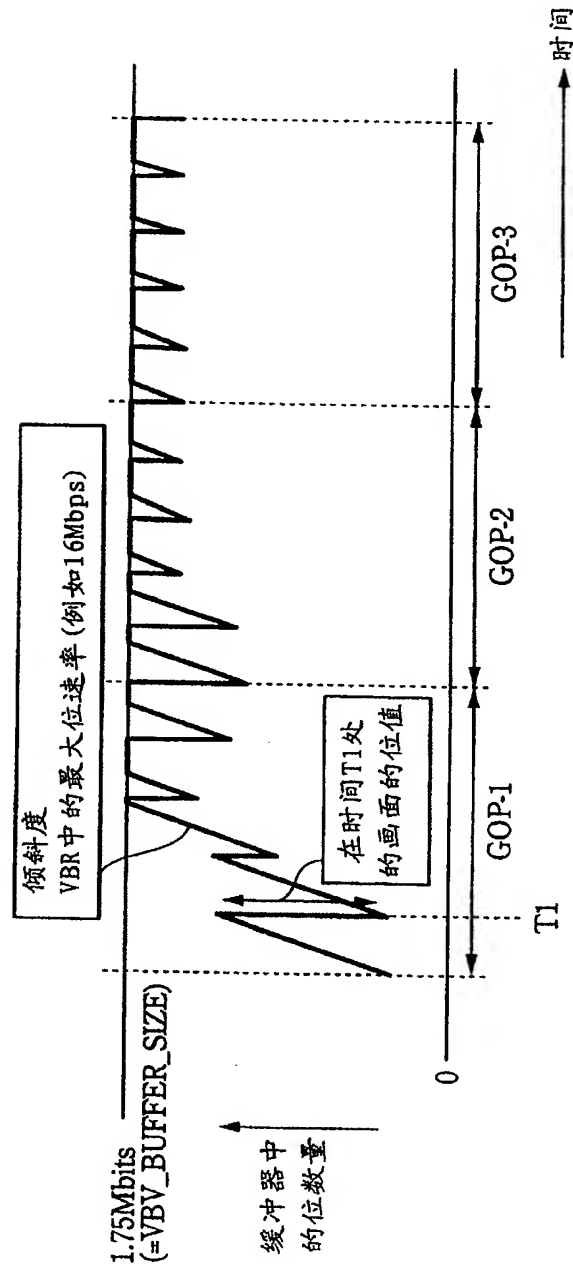


图 101



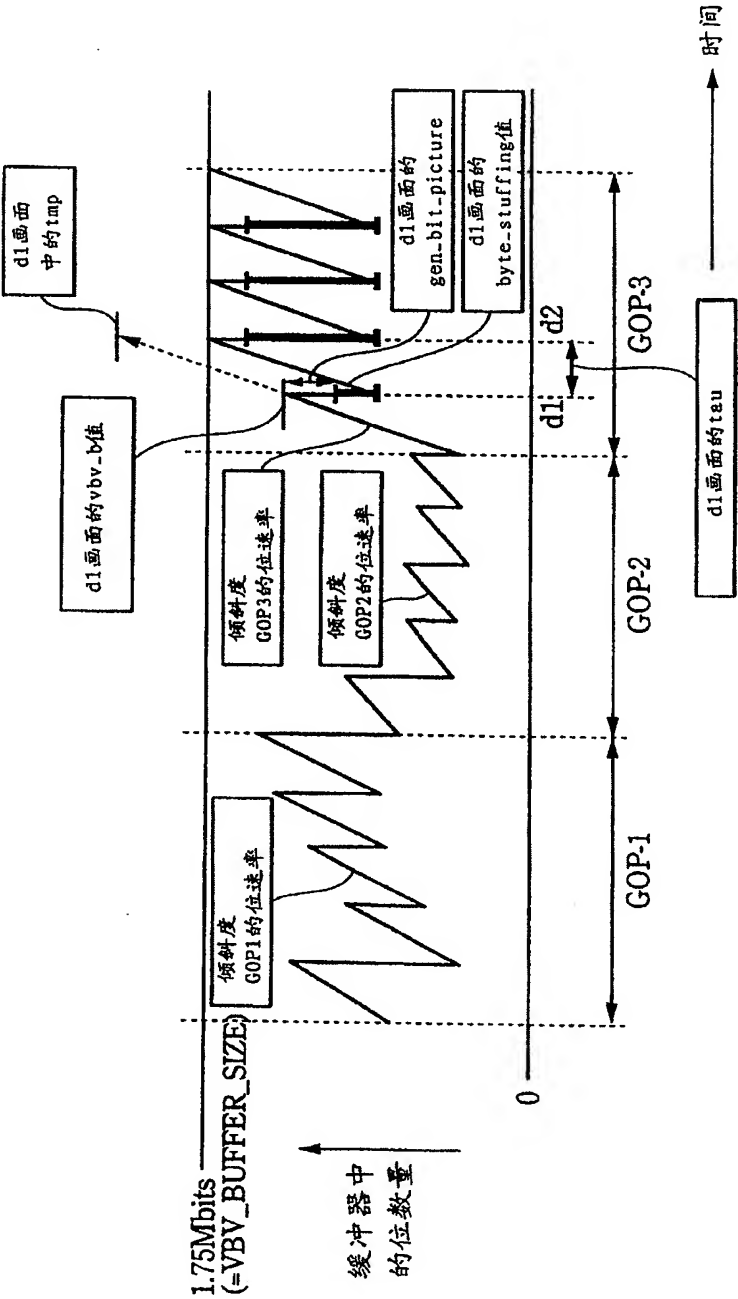


图 102

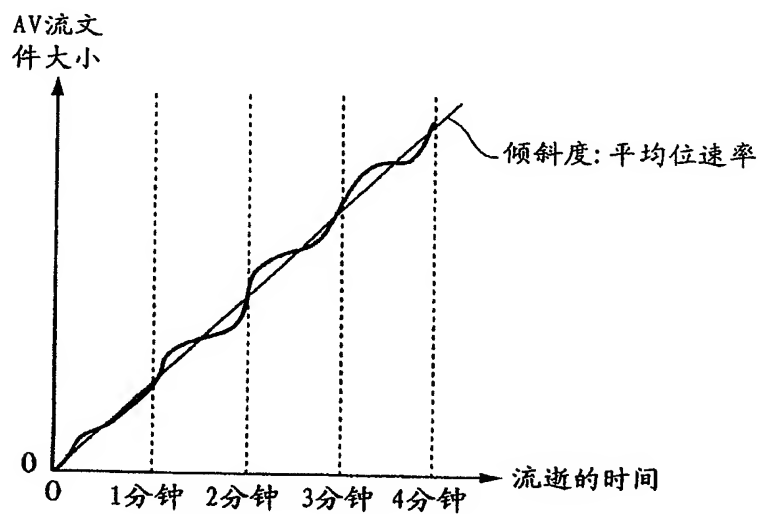


图 103

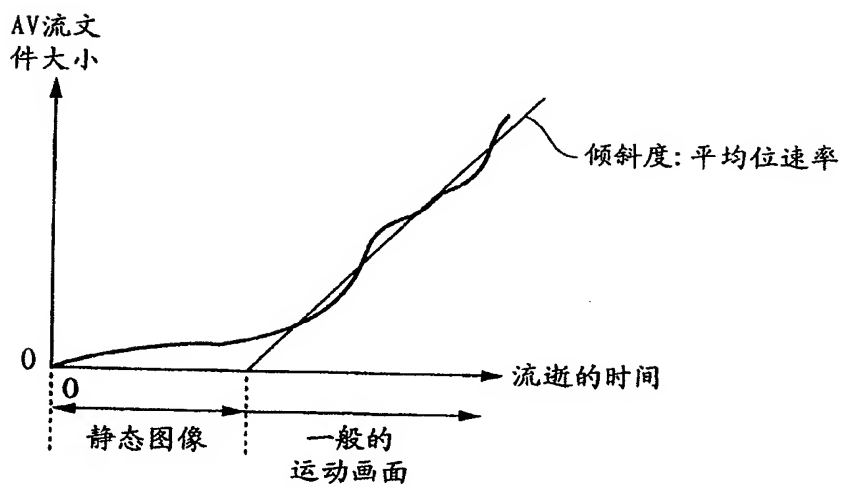


图 104

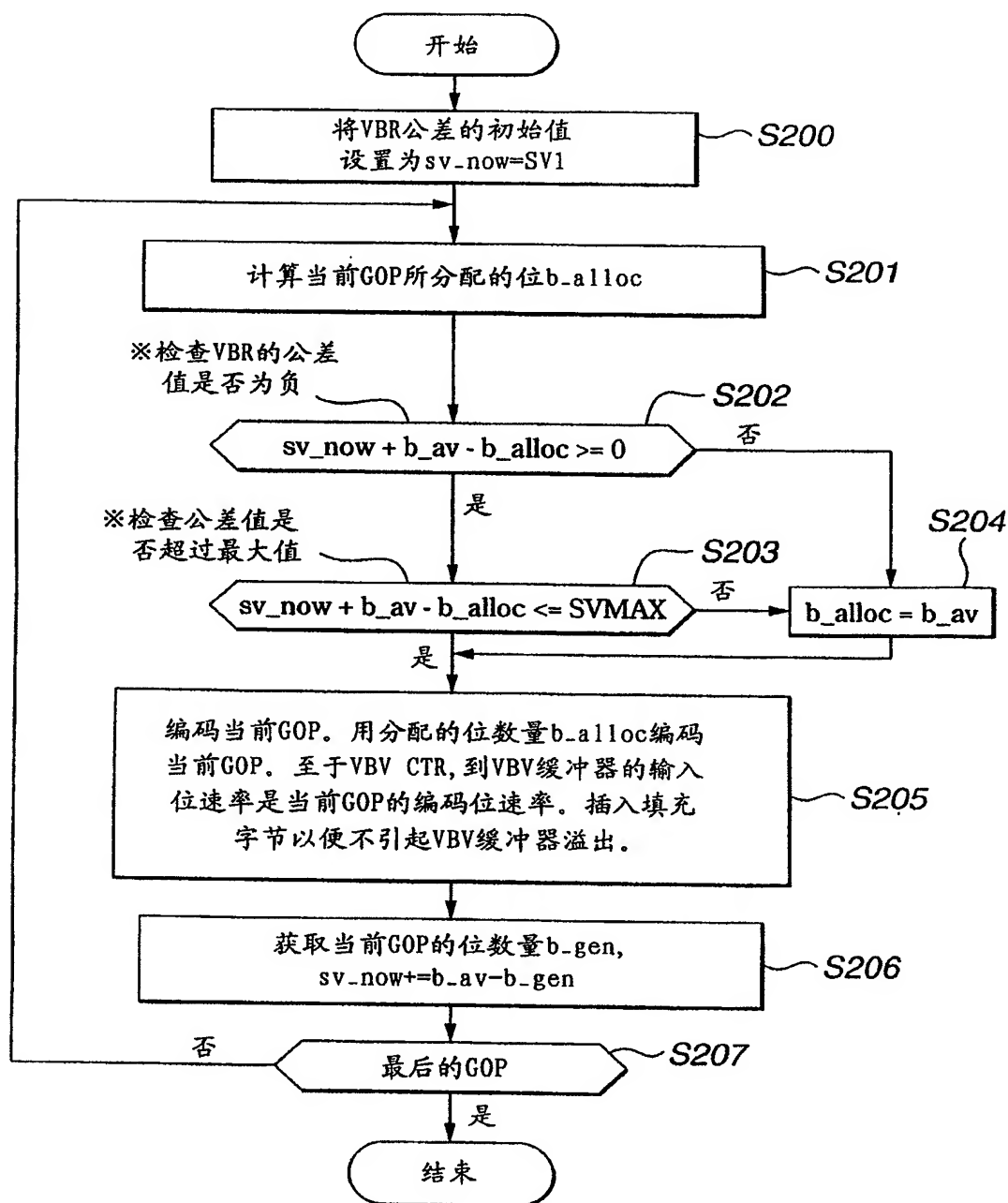


图 105

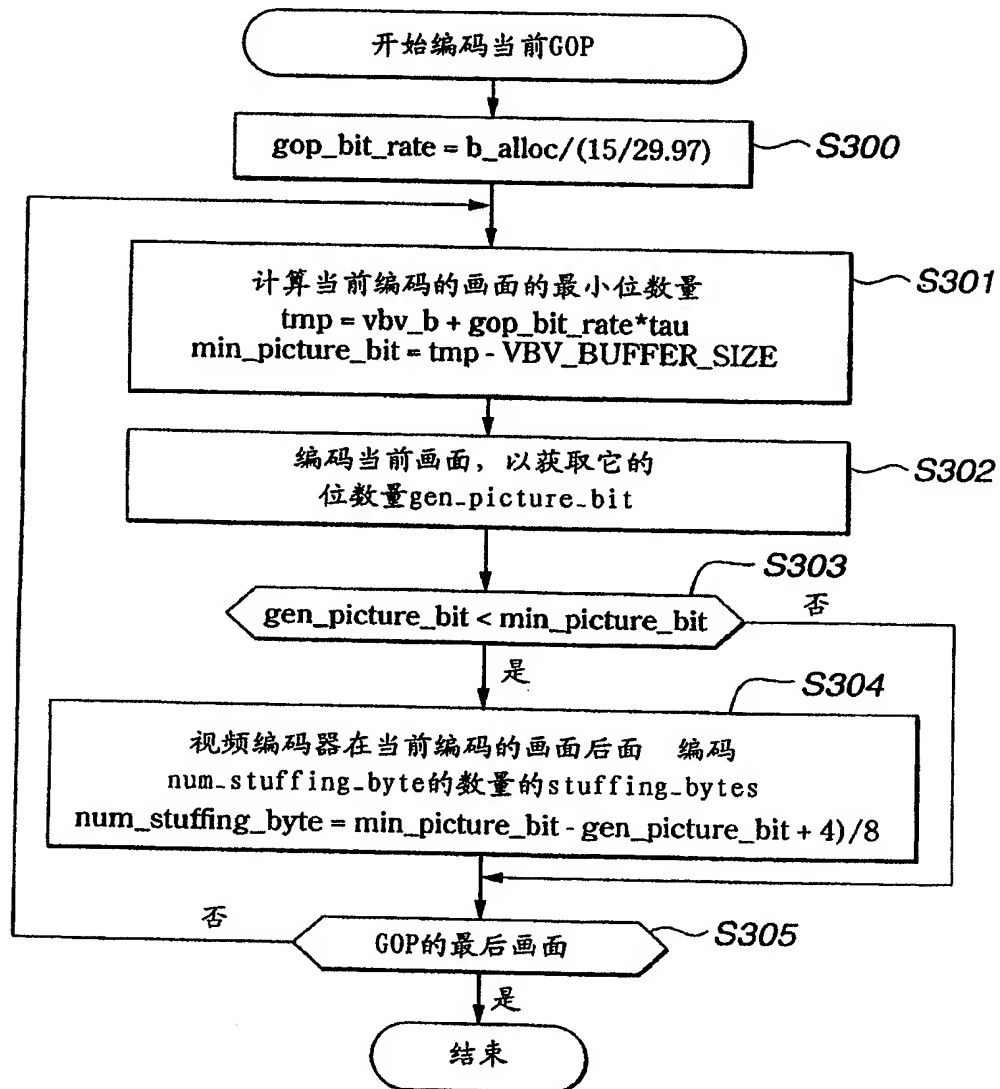


图 106

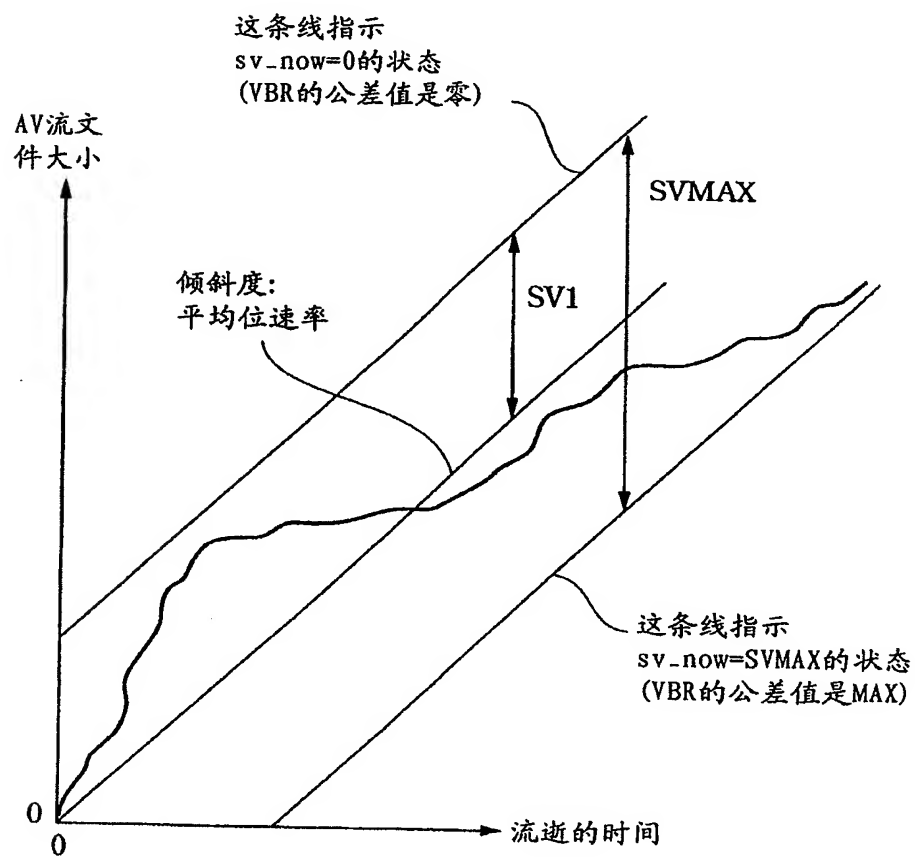


图 107

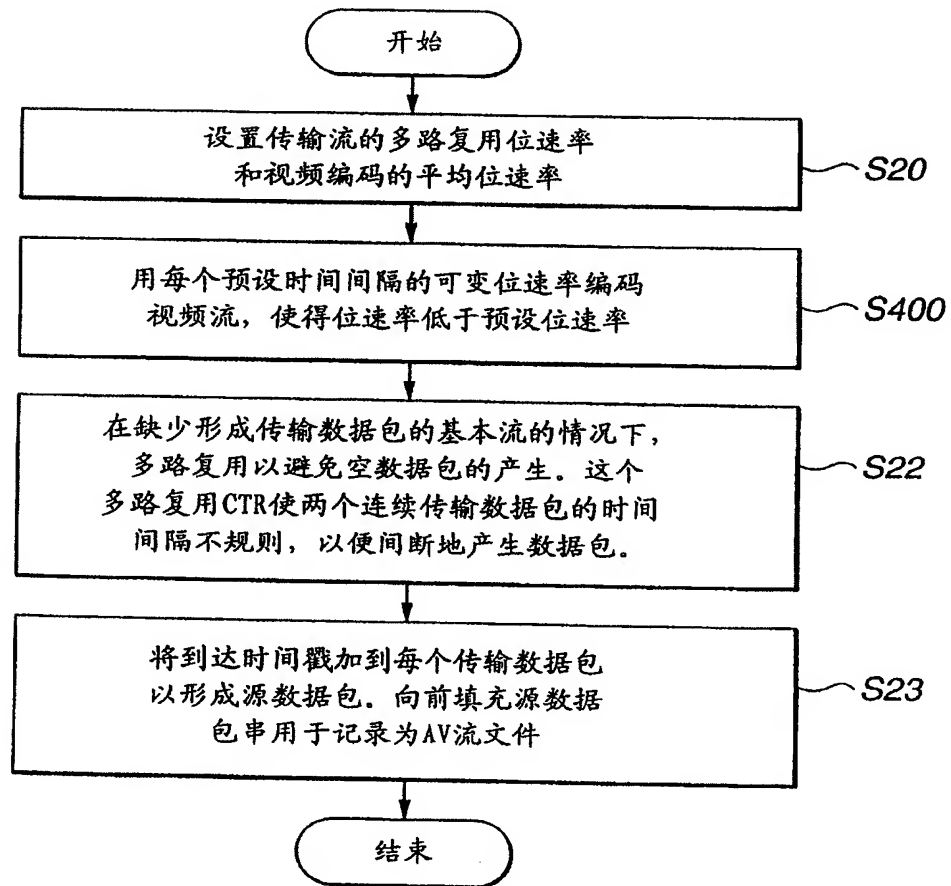


图 108

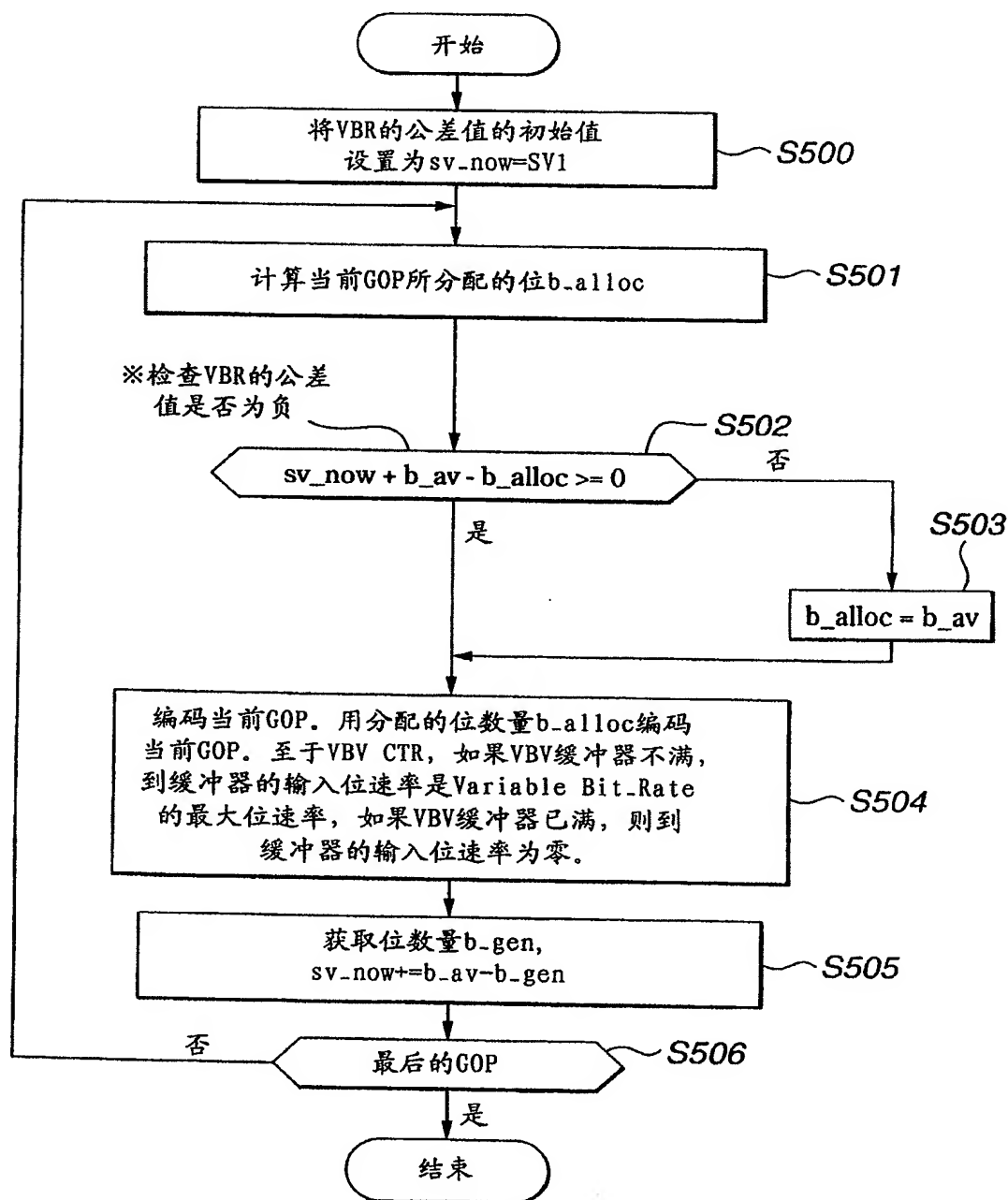


图 109

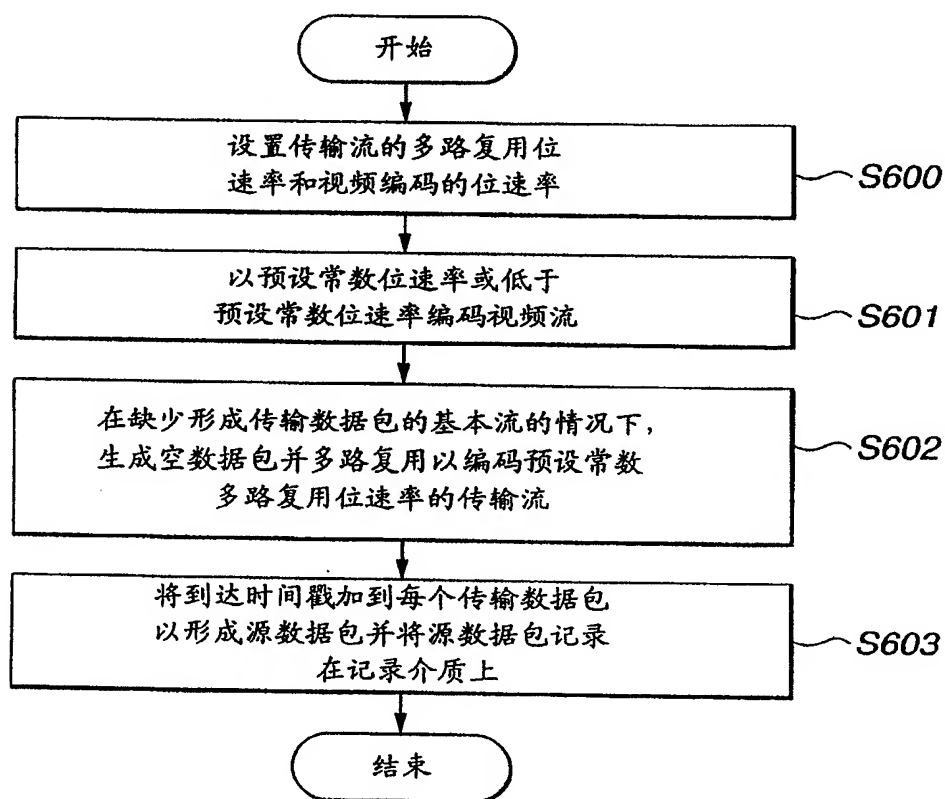


图 110



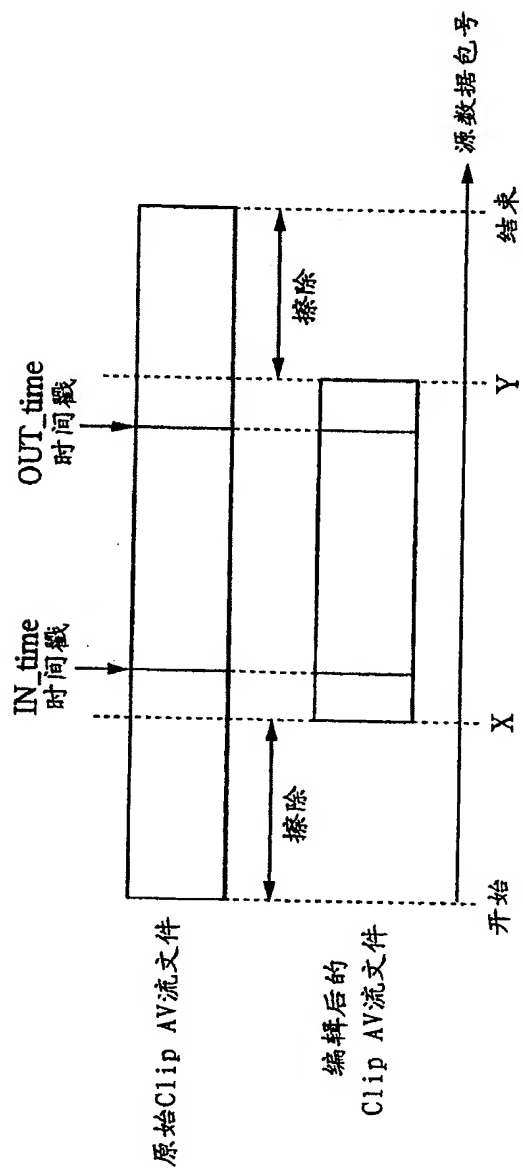


图 111

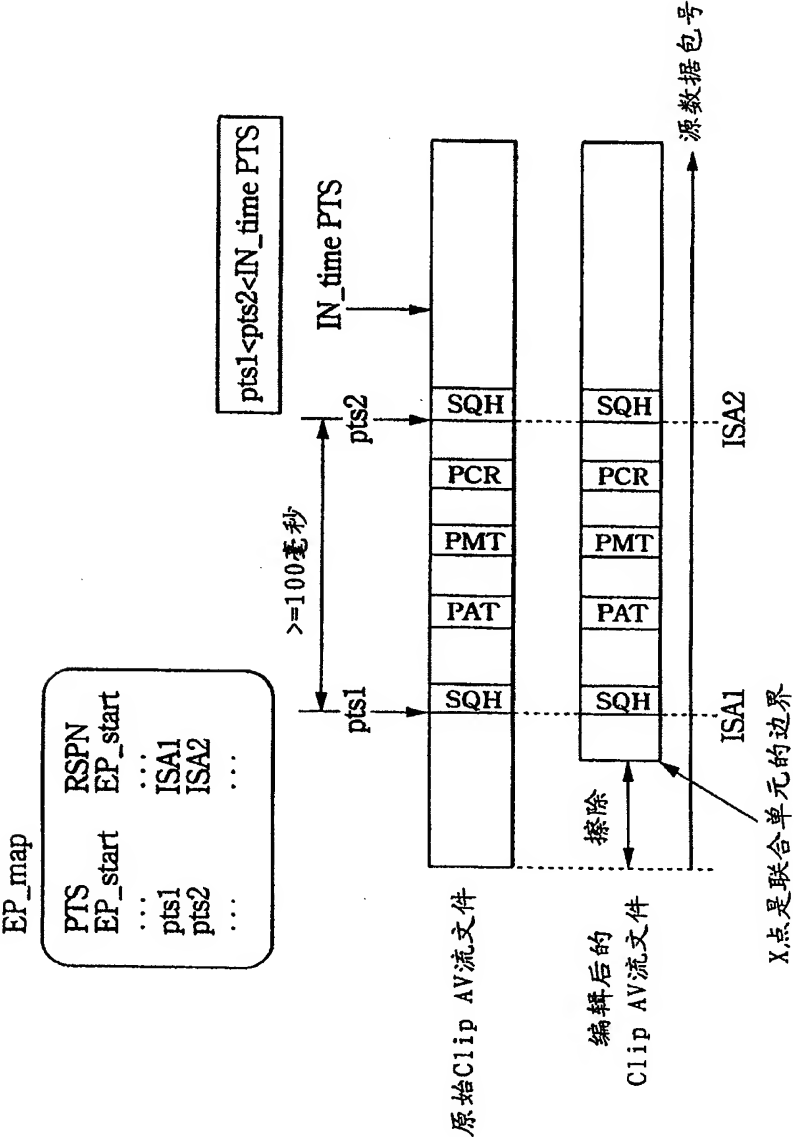


图 112

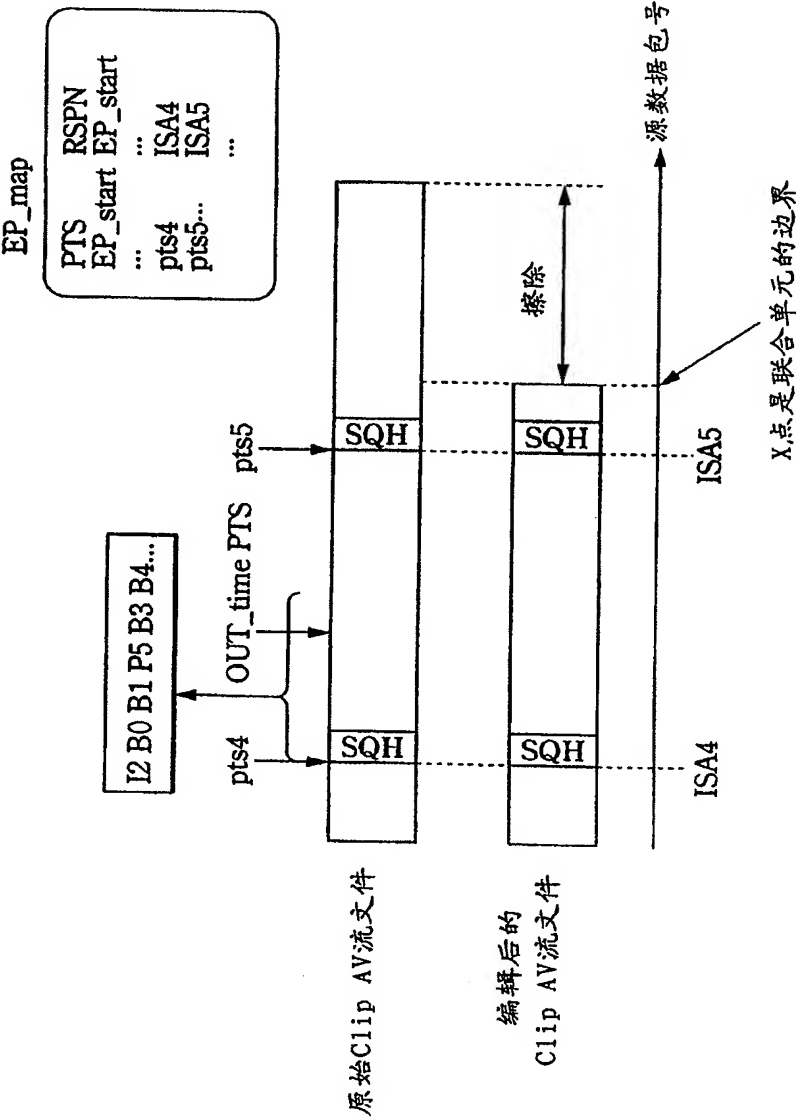


图 113

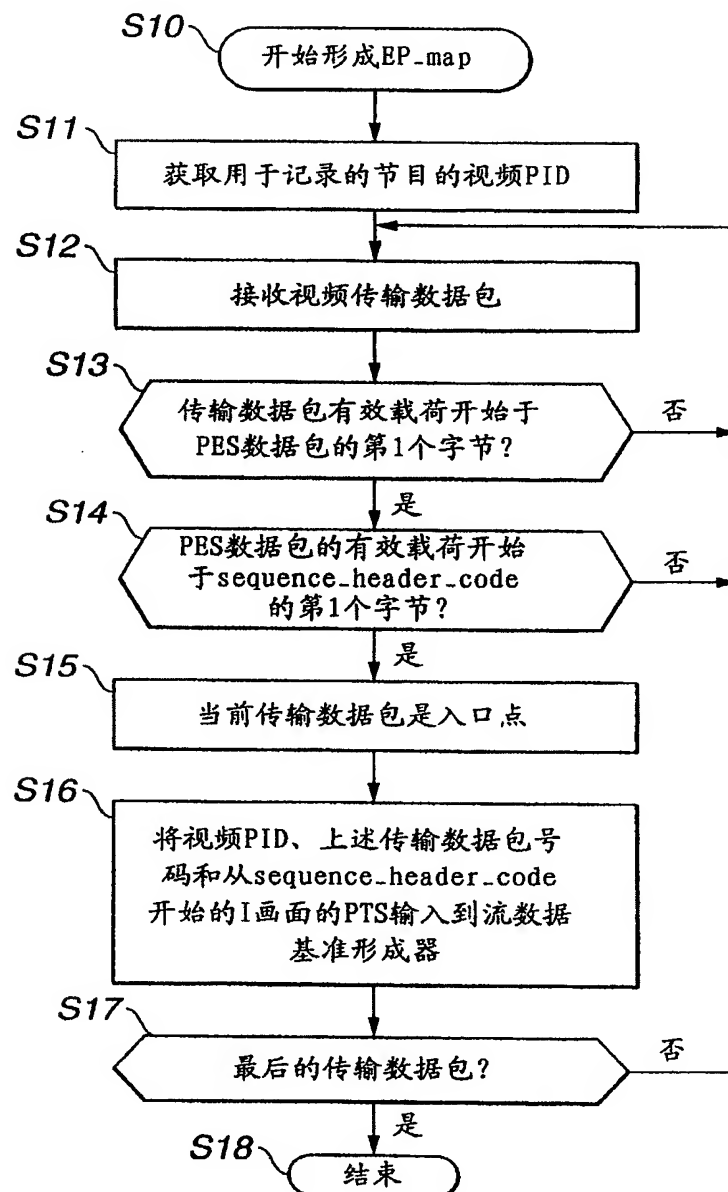


图 114

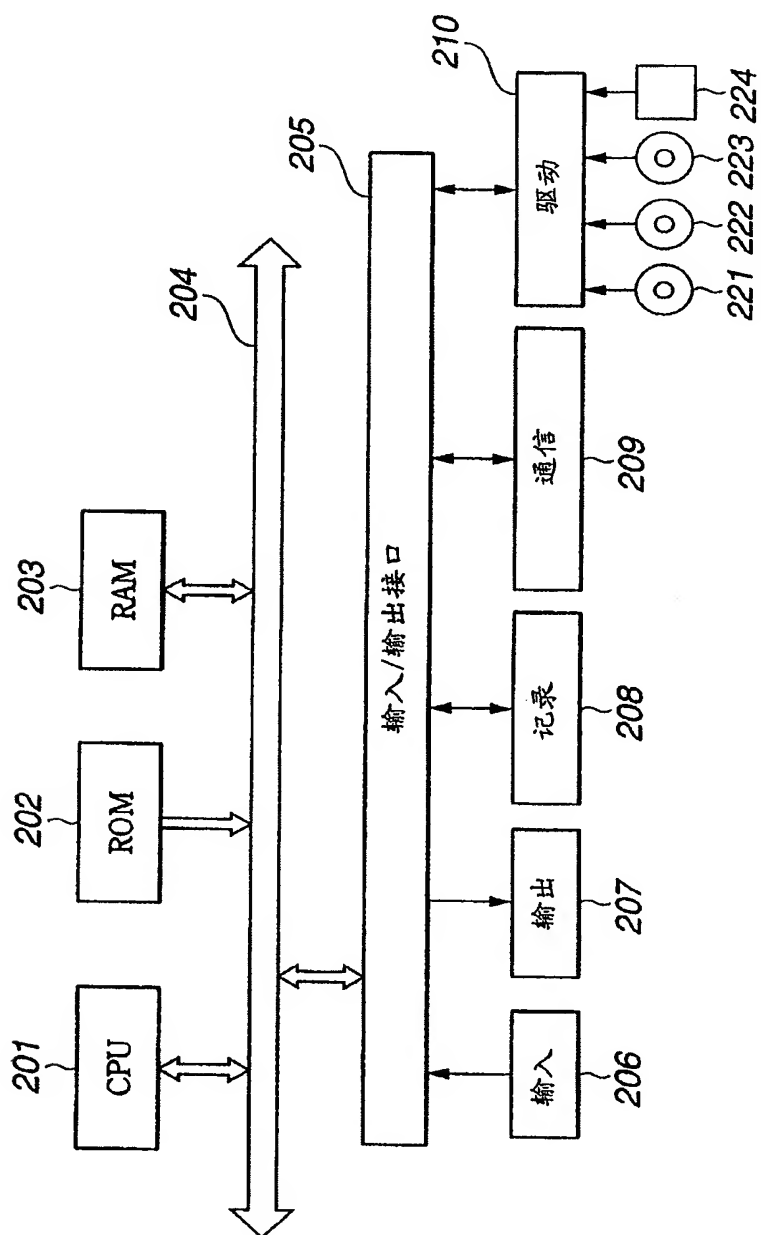


图 115